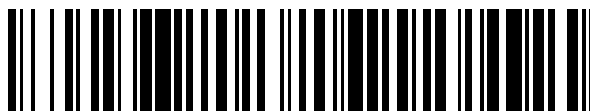


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 578 186**

51 Int. Cl.:

G06F 11/14 (2006.01)

G06F 12/16 (2006.01)

G06F 3/06 (2006.01)

G06F 9/44 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **03.03.2012 E 12757407 (7)**

97 Fecha y número de publicación de la concesión europea: **27.04.2016 EP 2684137**

54 Título: **Estrategias de copia de seguridad y de restauración para deduplicación de datos**

30 Prioridad:

11.03.2011 US 201113045692

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

21.07.2016

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)**

**One Microsoft Way
Redmond, WA 98052, US**

72 Inventor/es:

**KALACH, RAN;
CHEUNG, CHUN HO (IAN);
OLTEAN, PAUL ADRIAN y
DICKSON, MATHEW JAMES**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 578 186 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

Estrategias de copia de seguridad y de restauración para deduplicación de datos

Antecedentes

5 La deduplicación de datos, que también se conoce como optimización de datos, es el acto de reducir la cantidad física de bytes de datos que es necesario almacenar en disco o transmitir a través de una red sin poner en peligro la fidelidad o integridad de los datos originales. La deduplicación de datos reduce la capacidad de almacenamiento que se necesita para almacenar datos y, por lo tanto, puede conducir a ahorros en términos de costes de soporte físico de almacenamiento y de costes gestión de datos. La deduplicación de datos proporciona una solución al manejo del rápido crecimiento de los datos digitalmente almacenados.

10 La deduplicación de datos se puede realizar de acuerdo con una o más técnicas para eliminar la redundancia dentro de y entre archivos almacenados de forma persistente. Por ejemplo, de acuerdo con una técnica, se pueden identificar unas regiones únicas de datos que aparecen múltiples veces en uno o más archivos, y una única copia de esas regiones únicas identificadas de datos se puede almacenar físicamente. Se pueden almacenar referencias a esas regiones únicas identificadas de datos (a las que también se hace referencia como "fragmentos" de datos) que indican los archivos, y las ubicaciones en los archivos, que incluyen estos. Comúnmente, se hace referencia a esta técnica como instanciación única. Además de la instanciación única, se puede realizar la compresión de datos. Otras técnicas de reducción de datos también se pueden implementar como parte de una solución de deduplicación de datos.

20 Existen dificultades en la gestión de datos que se almacenan de acuerdo con las técnicas de deduplicación de datos. Por ejemplo, debido a la fragmentación de datos que es impuesta por la deduplicación de datos, puede existir latencia en el acceso a archivos que se almacenan de acuerdo con la deduplicación. Esta latencia limita la adopción de soluciones de deduplicación de datos, en especial en los datos de almacenamiento primario, en los que los usuarios esperan un acceso rápido y sin discontinuidades a los archivos. Además, los algoritmos de deduplicación de datos se pueden ejecutar en un aparato dedicado o en el dispositivo que almacena y sirve datos (por ejemplo, un servidor de archivos). En el caso de un servidor de archivos, puede que la deduplicación de datos no sea la función primaria del dispositivo y, por lo tanto, puede que sea necesario que las técnicas de deduplicación de datos sean eficientes con el fin de no consumir en exceso los recursos del dispositivo (por ejemplo, memoria, mecanismos de entrada/salida (E/S), capacidad de la unidad de procesamiento central (CPU, *central processing unit*), etc.). Lo que es más, debido a que la cantidad de datos digitales está creciendo a un ritmo muy alto, el tamaño de los dispositivos de almacenamiento (por ejemplo, discos de almacenamiento) y la capacidad de almacenamiento total que está asociada con los dispositivos informáticos está creciendo, dando lugar a dificultades con las técnicas de deduplicación de datos que no se adaptan bien a los cambios de escala con cantidades crecientes de almacenamiento. La deduplicación de datos posibilita, de hecho, que se realicen unas copias de seguridad de datos más pequeñas y unas copias de seguridad de datos más rápidas debido a la compresión de datos y, como resultado, posibilita unas restauraciones más rápidas de datos a los cuales se les ha realizado una copia de seguridad. No obstante, existen, de hecho, desafíos en la realización de una copia de seguridad de datos deduplicados y en la restauración de datos deduplicados a partir de un almacenamiento de copia de seguridad.

35 El documento US 2010/082672 A1 divulga un procedimiento para gestionar datos de instancia única. Usando una base de datos de instancia única, se mejora la densidad de datos en los medios de archivo. Para gestionar datos de instanciación única, para cada operación de almacenamiento, se crea una carpeta de fragmentos en un dispositivo de almacenamiento que almacena datos de instanciación única. La carpeta de fragmentos contiene tres archivos: 1) un archivo que contiene unos objetos de datos a los que se ha dado una instancia única; 2) un archivo que contiene unos objetos de datos que no han sido aptos para una instanciación única; y 3) un archivo de metadatos que se usa para realizar un seguimiento de la ubicación de los objetos de datos dentro de los otros archivos. Una segunda operación de almacenamiento posterior a una primera operación de almacenamiento contiene referencias a objetos de datos en la carpeta de fragmentos que es creada por la primera operación de almacenamiento en lugar de los propios objetos de datos.

40 El documento US 2009/177855 A1 divulga la realización de una copia de seguridad de un sistema de archivos informáticos deduplicado de un sistema informático, incluyendo dividir el sistema de archivos en particiones, almacenar cada una de las particiones en un medio de almacenamiento separado y mantener el estado deduplicado del sistema de archivos tanto como sea posible.

50 El documento US 2009/013129 A1 divulga la capacidad eficaz de un cartucho de datos para competir de cerca con la de un cartucho de cinta según la especificación *Linear Tape Open* (LTO). Esto se logra a través de la implementación de factorización de elementos en común (o deduplicación) en la HDD del cartucho de datos. Además, se divulgan unos procedimientos para optimizar el uso de factorización de elementos en común en situaciones de múltiples cartuchos.

55 El documento US 7870409 B2 divulga un procedimiento y sistema para reducir el consumo de energía en un sistema de copia de seguridad de datos mediante el control de fuente de alimentación de disco al tiempo que también se usa una función de deduplicación sobre el sistema para reducir la cantidad global de capacidad de almacenamiento que se requiere. El sistema ahorra energía al tiempo que también se da lugar a que el sistema de almacenamiento almacene datos de forma más eficiente mediante la realización de una función de deduplicación.

60 Por lo tanto, el objeto de la presente invención es la provisión de un procedimiento mejorado de copia de seguridad

de flujo de datos, un producto de programa informático correspondiente y un sistema correspondiente.

El presente objeto es solucionado por la invención, la cual se define con detalle por la materia objeto de las reivindicaciones independientes adjuntas 1, 9 y 10.

En las reivindicaciones dependientes se definen realizaciones preferidas.

5 **Sumario**

Este Sumario se proporciona para introducir una selección de conceptos en una forma simplificada que se describen adicionalmente en lo sucesivo en la Descripción Detallada. Este Sumario no tiene por objeto identificar rasgos distintivos clave o rasgos distintivos esenciales de la materia objeto que se reivindica, ni tiene por objeto usarse para limitar el ámbito de la materia objeto que se reivindica.

- 10 Se proporcionan procedimientos, sistemas y productos de programa informático para realizar una copia de seguridad de flujos de datos optimizados en un almacenamiento de copia de seguridad y para restaurar flujos de datos a partir de un almacenamiento de copia de seguridad. Datos optimizados en la presente memoria descriptiva se refieren a datos que se han optimizado, o desduplicado, mediante una o más de unas técnicas de desduplicación de datos tales como la instanciación única de fragmentos y la compresión. Los flujos optimizados se refieren a flujos
15 que se desduplicaron o, dicho de otra forma, sus datos se optimizaron usando técnicas de desduplicación de datos.

- Por ejemplo, se describen unas implementaciones para copia de seguridad de flujos de datos optimizados. Un almacén de fragmentos incluye una pluralidad de flujos de datos optimizados, con cada flujo de datos optimizado teniendo una pluralidad de fragmentos que incluye al menos un fragmento de datos y unos metadatos de flujo optimizado (por ejemplo, una asignación de flujo que se almacena como un fragmento de asignación de flujo en un
20 contenedor de fragmentos del almacén de fragmentos, una tabla global, una base de datos, etc.) que asignan unos identificadores para los fragmentos de datos con unas ubicaciones de los fragmentos de datos en el almacén de fragmentos. El almacén de fragmentos incluye fragmentos de datos de una forma desduplicada. Uno o más de los flujos de datos optimizados que están almacenados en el almacén de fragmentos se pueden identificar para copia de seguridad. Para realizar una copia de seguridad de los flujos de datos optimizados identificados, al menos una
25 porción del almacén de fragmentos se almacena en un almacenamiento de copia de seguridad. El almacén de fragmentos (una porción del mismo, o su totalidad) se almacena en el almacenamiento de copia de seguridad de acuerdo con una técnica de copia de seguridad seleccionada.

- Por ejemplo, de acuerdo con una técnica de copia de seguridad optimizada, uno o más contenedores de fragmentos del almacén de fragmentos se almacenan en el almacenamiento de copia de seguridad en su totalidad. Una
30 pluralidad de estructuras de flujo optimizado se almacenan en el almacenamiento de copia de seguridad. Las estructuras de flujo optimizado son puntos de repetición de análisis sintáctico para los flujos de datos optimizados identificados para copia de seguridad.

- De acuerdo con una técnica de copia de seguridad no optimizada, cada flujo de datos optimizado identificado para copia de seguridad se rehidrata en un flujo de datos no optimizado correspondiente que incluye cualesquiera
35 fragmentos de datos a los que hacen referencia los metadatos del flujo de datos optimizado. Cada flujo de datos no optimizado se almacena en el almacenamiento de copia de seguridad.

- De acuerdo con una técnica de copia de seguridad a nivel de elemento, un primer flujo de datos optimizado se identifica para copia de seguridad. Se determinan uno o más fragmentos de datos a los que hacen referencia los metadatos del primer flujo de datos optimizado que no están almacenados aún en el almacenamiento de copia de
40 seguridad. Los metadatos de flujo optimizado del primer flujo de datos optimizado se almacenan en el almacenamiento de copia de seguridad. Los fragmentos de datos que se determina que no se les ha realizado aún una copia de seguridad también se almacenan en el almacenamiento de copia de seguridad. Cada flujo de datos optimizado identificado para copia de seguridad se puede almacenar en un almacenamiento de copia de seguridad de una forma similar.

- De acuerdo con una técnica de copia de seguridad de identificadores de fragmento de datos, los metadatos de flujo optimizado de cada flujo de datos optimizado se analizan para determinar identificadores de fragmento de datos para los fragmentos de datos a los que hacen referencia los metadatos de flujo optimizado. Una estructura de flujo optimizado para cada flujo de datos optimizado se almacena en el almacenamiento de copia de seguridad con el al
45 menos un identificador de fragmento de datos correspondiente. Uno o más contenedores de fragmentos del almacén de fragmentos que almacenan los fragmentos de datos a los que se hace referencia también se almacenan en el
50 almacenamiento de copia de seguridad.

Una técnica de copia de seguridad se puede seleccionar basándose en heurística. La técnica de copia de seguridad seleccionada se realiza para realizar una copia de seguridad de la pluralidad de flujos de datos optimizados en el almacenamiento de copia de seguridad.

- 55 También se describen unas implementaciones para la restauración de flujos de datos optimizados a partir de un almacenamiento de copia de seguridad. Se recibe una solicitud para que un flujo de datos optimizado se recupere de un almacén de fragmentos en un almacenamiento de copia de seguridad. La solicitud incluye un identificador para metadatos de flujo optimizado que se corresponden con el flujo de datos. Una primera llamada a una aplicación

de restauración se genera basándose en el identificador de metadatos de flujo optimizado. La primera llamada específica un nombre de archivo para un contenedor de flujo en un almacenamiento de copia de seguridad que almacena los metadatos de flujo optimizado que son identificados por el identificador de metadatos de flujo optimizado, y especifica un desplazamiento para los metadatos de flujo optimizado en el contenedor de flujo. Los metadatos de flujo optimizado se reciben en respuesta a la primera llamada. Se identifica al menos un identificador de fragmento de datos al que se hace referencia en los metadatos de flujo optimizado. Se genera al menos una llamada adicional a la aplicación de restauración en correspondencia con el identificador o identificadores de fragmento de datos a los que hacen referencia los metadatos de flujo optimizado para obtener al menos un fragmento de datos a partir de al menos un contenedor de fragmentos en un almacenamiento de copia de seguridad. El fragmento o fragmentos de datos se recibe o reciben en respuesta a la llamada o llamadas adicionales. El fragmento o fragmentos de datos son combinables para restaurar el flujo de datos.

En una implementación, los identificadores de fragmento de datos a los que hacen referencia metadatos de flujo optimizado se pueden clasificar basándose en un contenedor de fragmentos como una primera clave de clasificación y un desplazamiento de fragmento como una segunda clave. Las llamadas a la aplicación de restauración para recuperar fragmentos de datos se pueden generar en un orden que es definido por la clasificación.

En una implementación, un motor de restauración puede llamar a la aplicación de restauración con un plan de restauración, el cual puede incluir una solicitud para determinar el orden de la restauración mediante la clasificación de los archivos de copia de seguridad que contienen fragmentos de datos, o para informar a la aplicación de restauración acerca de qué extensiones de archivos están a punto de restaurarse, de tal modo que la aplicación de restauración puede optimizar mediante la realización de unas operaciones de almacenamiento en memoria caché y de lectura anticipada.

Se describen unas implementaciones de sistemas para copia de seguridad y/o la restauración de flujos de datos optimizados. En una implementación, un módulo de copia de seguridad de datos recibe una identificación de una pluralidad de flujos de datos optimizados almacenados en un almacén de fragmentos al que se va a realizar una copia de seguridad. El módulo de copia de seguridad de datos está configurado para almacenar al menos una porción del almacén de fragmentos en un almacenamiento de copia de seguridad para realizar una copia de seguridad de la pluralidad de flujos de datos optimizados.

En unas implementaciones, el módulo de copia de seguridad de datos puede incluir uno o más de un módulo de copia de seguridad de archivos optimizada, un módulo de copia de seguridad de rehidratación, un módulo de copia de seguridad a nivel de elemento y/o un módulo de copia de seguridad de identificadores de fragmento de datos. El módulo de copia de seguridad de archivos optimizada está configurado para almacenar el almacén de fragmentos en su totalidad en el almacenamiento de copia de seguridad así como estructuras de datos optimizadas como puntos de repetición de análisis sintáctico para los flujos de datos optimizados. El módulo de copia de seguridad de rehidratación está configurado para rehidratar cada flujo de datos optimizado en un flujo de datos no optimizado correspondiente que incluye cualesquiera fragmentos de datos a los que hacen referencia los metadatos de flujo optimizado. El módulo de copia de seguridad de rehidratación almacena cada flujo de datos no optimizado en el almacenamiento de copia de seguridad. El módulo de copia de seguridad a nivel de elemento está configurado para cada flujo de datos optimizado para determinar uno o más fragmentos de datos a los que hacen referencia metadatos de los metadatos de flujo optimizado del flujo de datos optimizado que no están almacenados aún en el almacenamiento de copia de seguridad. El módulo de copia de seguridad a nivel de elemento almacena los metadatos de flujo optimizado del flujo de datos optimizado en el almacenamiento de copia de seguridad, y almacena uno o más fragmentos de datos que se determina que son duplicados en el almacenamiento de copia de seguridad. El módulo de copia de seguridad de identificadores de fragmento de datos está configurado para analizar los metadatos de flujo optimizado de cada flujo de datos optimizado para determinar al menos un identificador de fragmento de datos para el fragmento o fragmentos de datos a los que hacen referencia los metadatos de flujo optimizado. El módulo de copia de seguridad de identificadores de fragmento de datos almacena cada flujo de datos optimizado en el almacenamiento de copia de seguridad con el identificador o identificadores de fragmento de datos correspondientes, y almacena uno o más contenedores de fragmentos del almacén de fragmentos en el almacenamiento de copia de seguridad.

En una implementación, el módulo de copia de seguridad de datos puede incluir un módulo de heurística. El módulo de heurística está configurado para seleccionar una técnica de copia de seguridad basándose en heurística, incluyendo una heurística tal como los tamaños de los flujos de datos a los que se va a realizar una copia de seguridad, los tamaños de los flujos de datos después de ser deduplicados (para eliminar fragmentos de datos redundantes), el tamaño global del almacén de fragmentos, y una heurística adicional, tal como una heurística que se puede generar basándose en estas cantidades. El módulo de heurística posibilita que la técnica de copia de seguridad se realice para realizar una copia de seguridad de la pluralidad de flujos de datos optimizados en el almacenamiento de copia de seguridad.

En otra implementación, un módulo de restauración de datos está configurado para restaurar flujos de datos optimizados a partir de un almacenamiento de copia de seguridad. El módulo de restauración de datos incluye un reconstructor de archivos que recibe una solicitud para que un flujo de datos se recupere de un almacén de fragmentos en el almacenamiento de copia de seguridad. La solicitud incluye un identificador para metadatos de flujo

optimizado que se corresponden con el flujo de datos. El reconstructor de archivos incluye un módulo de devolución de llamada (por ejemplo, proporcionado por un sistema de copia de seguridad/restauración). El módulo de devolución de llamada genera una primera llamada a una aplicación de restauración basándose en el identificador de metadatos de flujo optimizado. La primera llamada especifica un nombre de archivo para un primer contenedor de fragmentos en el almacenamiento de copia de seguridad que almacena metadatos de flujo optimizado identificados por el identificador de metadatos de flujo optimizado, y especifica un desplazamiento para los metadatos de flujo optimizado en el primer contenedor de fragmentos. El reconstructor de archivos recibe los metadatos de flujo optimizado en respuesta a la primera llamada, y determina al menos un identificador de fragmento de datos al que se hace referencia en los metadatos de flujo optimizado. El módulo de devolución de llamada genera al menos una llamada adicional a la aplicación de restauración que se corresponde con el identificador o identificadores de fragmento de datos para obtener uno o más fragmentos de datos a partir de uno o más contenedores de fragmentos en el almacenamiento de copia de seguridad. El reconstructor de archivos recibe el fragmento o fragmentos de datos en respuesta a la llamada o llamadas adicionales, y puede combinar el fragmento o fragmentos de datos para restaurar el flujo de datos. Como alternativa, el constructor de archivos puede almacenar los metadatos de flujo optimizado y el fragmento o fragmentos de datos en un almacén de fragmentos, y una estructura de flujo optimizado para el flujo de datos se puede almacenar en un almacenamiento que hace referencia a los fragmentos que se almacenan en el almacén de fragmentos.

El módulo de devolución de llamada también puede llamar a la aplicación de restauración con un plan de restauración, el cual puede incluir una solicitud para determinar el orden de la restauración mediante la clasificación de la lista de archivos de contenedor de fragmentos de datos, o para informar a la aplicación de restauración acerca de qué extensiones están a punto de restaurarse a partir de los archivos de contenedor, de tal modo que la aplicación de restauración puede optimizar mediante la realización de unas operaciones de almacenamiento en memoria caché y de lectura anticipada.

También se describen productos de programa informático en el presente documento para realizar una copia de seguridad de flujos de datos optimizados en un almacenamiento de copia de seguridad, para restaurar flujos de datos optimizados a partir de un almacenamiento de copia de seguridad, y para realizaciones adicionales tal como se describe en el presente documento.

Rasgos distintivos y ventajas adicionales de la invención, así como la estructura y el funcionamiento de varias realizaciones de la invención, se describen con detalle en lo sucesivo con referencia a los dibujos adjuntos. Se hace notar que la invención no se limita a las realizaciones específicas que se describen en el presente documento. Tales realizaciones se presentan en el presente documento solo por razones de ilustración. Realizaciones adicionales resultarán evidentes a los expertos en la materia o materias relevantes basándose en las enseñanzas contenidas en el presente documento.

Breve descripción de los dibujos/figuras

Los dibujos adjuntos, los cuales se incorporan en el presente documento y forman una parte de la memoria descriptiva, ilustran la presente invención y, junto con la descripción, sirven adicionalmente para explicar los principios de la invención y para posibilitar que un experto en la materia pertinente realice y use la invención.

La figura 1 muestra un diagrama de bloques de un sistema de deduplicación de datos, de acuerdo con una realización a modo de ejemplo.

La figura 2 muestra un diagrama de bloques de un almacén de fragmentos, de acuerdo con una realización a modo de ejemplo.

La figura 3 muestra un diagrama de bloques de un almacén de fragmentos, de acuerdo con una realización a modo de ejemplo.

La figura 4 muestra un diagrama de bloques de un sistema de almacén de flujo de datos, de acuerdo con una realización a modo de ejemplo.

La figura 5 muestra un diagrama de flujo para almacenar un flujo de datos, de acuerdo con una realización a modo de ejemplo.

La figura 6 muestra un diagrama de bloques que ilustra un ejemplo del almacenamiento de flujos de datos en un almacén de datos, de acuerdo con una realización.

La figura 7 muestra un diagrama de bloques de una interfaz de almacén de fragmentos que incluye un módulo de rehidratación, de acuerdo con una realización a modo de ejemplo.

La figura 8 muestra un diagrama de bloques de un contenedor de fragmentos, de acuerdo con una realización a modo de ejemplo.

La figura 9 muestra un diagrama de bloques de una tabla de redirección, de acuerdo con una realización a modo de ejemplo.

La figura 10 muestra un diagrama de flujo para almacenar un flujo de datos, de acuerdo con una realización a modo de ejemplo.

La figura 11 muestra un diagrama de bloques de un módulo de rehidratación que accede a un almacén de fragmentos para rehidratar un flujo de datos, de acuerdo con una realización a modo de ejemplo.

La figura 12 muestra un diagrama de flujo para realizar una copia de seguridad de datos en un sistema de optimización de datos, de acuerdo con una realización a modo de ejemplo.

La figura 13 muestra un diagrama de bloques de un sistema de copia de seguridad de datos, de acuerdo con una realización a modo de ejemplo.

Las figuras 14 - 17 muestran unos diagramas de flujo para realizar varias técnicas de copia de seguridad para flujos de datos optimizados, de acuerdo con realizaciones a modo de ejemplo.

La figura 18 muestra un diagrama de flujo que proporciona un procedimiento para seleccionar y ejecutar una técnica de copia de seguridad basándose en heurística, de acuerdo con una realización a modo de ejemplo.

La figura 19 muestra un diagrama de flujo para realizar una restauración de datos de copia de seguridad en un sistema de optimización de datos, de acuerdo con una realización a modo de ejemplo.

La figura 20 muestra un diagrama de bloques de un sistema de restauración de datos, de acuerdo con una realización a modo de ejemplo.

La figura 21 muestra un diagrama de bloques de un módulo de devolución de llamada que accede a una tabla de redistribución, de acuerdo con una realización.

La figura 22 muestra un diagrama de flujo que proporciona un procedimiento para acceder a una tabla de redistribución para obtener desplazamientos actualizados para los fragmentos, de acuerdo con una realización a modo de ejemplo.

La figura 23 muestra un diagrama de bloques de un reconstructor de archivos, de acuerdo con una realización a modo de ejemplo.

La figura 24 muestra un diagrama de flujo que proporciona un procedimiento para clasificar un orden de restauración de fragmento de datos, de acuerdo con una realización a modo de ejemplo.

La figura 25 muestra un diagrama de bloques de un ordenador a modo de ejemplo que se puede usar para implementar realizaciones de la presente invención.

Los rasgos distintivos y ventajas de la presente invención se volverán más evidentes a partir de la descripción detallada que se expone en lo sucesivo cuando se tome junto con los dibujos, en los que caracteres de referencia semejantes identifican elementos correspondientes por la totalidad de los mismos. En los dibujos, números de referencia semejantes indican en general unos elementos idénticos, funcionalmente similares y/o estructuralmente similares. El dibujo en el que un elemento aparece por primera vez se indica mediante el dígito o dígitos más a la izquierda en el número de referencia correspondiente.

Descripción detallada

I. Introducción

La presente memoria descriptiva divulga una o más realizaciones que incorporan los rasgos distintivos de la invención. La realización o realizaciones que se divulgan meramente ejemplifican la invención. El ámbito de la invención no se limita a la realización o realizaciones que se divulgan. La invención se define mediante las reivindicaciones que se adjuntan a la misma.

Las referencias en la memoria descriptiva a “una realización”, “una realización”, “una realización a modo de ejemplo”, etc., indican que la realización que se describe puede incluir un rasgo distintivo, estructura o característica particular, pero puede que no todas las realizaciones incluyan necesariamente el rasgo distintivo, estructura o característica particular. Además, tales expresiones no están haciendo referencia necesariamente a la misma realización. Además, cuando un rasgo distintivo, estructura o característica particular se describe en conexión con una realización, se afirma que se encuentra dentro del conocimiento de un experto en la materia el efectuar tal rasgo distintivo, estructura o característica en conexión con otras realizaciones, ya se describan de forma explícita o no.

Datos optimizados en la presente memoria descriptiva se refieren a datos que se han optimizado, o desduplicado, mediante una o más de unas técnicas de desduplicación de datos tales como la instanciación única de fragmentos y la compresión. Los flujos optimizados se refieren a flujos que se desduplicaron o, dicho de otra forma, sus datos se optimizaron usando técnicas de desduplicación de datos.

II. Realizaciones a modo de ejemplo

Unas realizaciones proporcionan técnicas para la desduplicación de datos. Tales realizaciones posibilitan que la cantidad de datos (por ejemplo, número de bytes) se almacene, o que se transmita, para reducirse sin poner en peligro la fidelidad o integridad de los datos. Por ejemplo, unas realizaciones posibilitan reducciones en la cantidad de latencia en el acceso a datos optimizados. Además, unas realizaciones posibilitan que los recursos, tales como máquinas/dispositivos informáticos, se usen de forma más eficiente, reduciendo el consumo de recursos. Lo que es más, unas realizaciones proporcionan técnicas para la desduplicación de datos, para la copia de seguridad de datos desduplicados, y para la restauración de datos a los cuales se les ha realizado una copia de seguridad a partir de almacenamiento que son escalonables con el crecimiento de la cantidad de datos digitales que se almacenan.

Por ejemplo, en una realización, se proporciona un almacén de fragmentos escalonable para la desduplicación de datos. El almacén de fragmentos posibilita varias técnicas para reducir al mínimo la latencia en un acceso a datos optimizados, reducir el consumo de recursos de máquina (por ejemplo, E/S de disco y memoria) y potenciar la fiabilidad durante la desduplicación de datos, la realización de una copia de seguridad de datos y la restauración de datos. En las siguientes subsecciones se describen realizaciones a modo de ejemplo con detalle adicional.

A. Realizaciones de deduplicación de datos a modo de ejemplo

En unas realizaciones, unos datos que se van a almacenar se pueden optimizar para reducir una cantidad de almacenamiento que es necesaria para los datos. Por ejemplo, los flujos de datos se pueden almacenar en forma de fragmentos de datos únicos. Se puede hacer referencia a los fragmentos de datos mediante unos metadatos de flujo optimizado que definen los flujos de datos. De esta forma, los flujos de datos se almacenan de forma más eficiente, debido a que los metadatos para múltiples flujos de datos pueden hacer referencia al mismo fragmento de datos almacenado, en lugar de que el mismo fragmento de datos se almacene múltiples veces. Además, los datos optimizados se pueden solicitar (por ejemplo, por aplicaciones) del almacenamiento según se desee. En tal caso, los flujos de datos se pueden volver a ensamblar a partir de los fragmentos de datos almacenados de acuerdo con los metadatos correspondientes.

Por ejemplo, la figura 1 muestra un diagrama de bloques de un sistema de deduplicación de datos 100, de acuerdo con una realización a modo de ejemplo. Tal como se muestra en la figura 1, el sistema 100 incluye un sistema de almacenamiento 102, un módulo de deduplicación de datos 104, un módulo de mantenimiento 106 y un almacenamiento 108. Además, el sistema de almacenamiento 102 incluye una API (*application programming interface*, interfaz de programación de aplicaciones) de flujo de datos 110, una API de mantenimiento de fragmentos 112 y una API de acceso de datos 114. El sistema 100 se describe tal como sigue para ilustrar el almacenamiento de datos optimizados, y la recuperación de datos optimizados a partir del almacenamiento, y no tiene por objeto ser limitante.

El sistema 100 está configurado para posibilitar que se almacenen datos en el almacenamiento 108 de una forma eficiente, y para que se recuperen datos del almacenamiento 108. Por ejemplo, en una realización, el módulo de deduplicación de datos 104 se puede encontrar presente. El módulo de deduplicación de datos 104 está configurado para optimizar los datos recibidos para su almacenamiento. Por ejemplo, el módulo de deduplicación de datos 104 puede comprimir datos recibidos que se reciben como un flujo de datos 132. El flujo de datos 132 puede incluir una porción de un archivo de datos, un único archivo de datos, múltiples archivos de datos, y/o cualquier combinación de archivos y/o porciones de archivos. Tal como se muestra en la figura 1, el módulo de deduplicación de datos 104 genera los fragmentos de datos 124, los cuales pueden ser una versión comprimida y segmentada del flujo de datos 132.

La API de flujo de datos 110 proporciona una interfaz para que el sistema de almacenamiento 102 reciba los fragmentos de datos 124. Los fragmentos de datos 124 pueden incluir una pluralidad de fragmentos de datos que forman el flujo de datos 132 a partir del cual se generan los fragmentos de datos 124. La API de flujo de datos 110 se puede configurar de cualquier forma adecuada, tal como sería conocido por los expertos en la materia o materias relevantes. La API de flujo de datos 110 puede emitir los fragmentos de datos 124 para que sean recibidos por la interfaz de almacén de fragmentos 116.

Tal como se muestra en la figura 1, el almacenamiento 108 está acoplado con el sistema de almacenamiento 102. La interfaz de almacén de fragmentos 116 es una interfaz entre las API 110, 112 y 114 y el almacenamiento 108. Por ejemplo, la interfaz de almacén de fragmentos 116 puede recibir los fragmentos de datos 124 y puede almacenar los fragmentos de datos de los fragmentos de datos 124 en el almacenamiento 108. Por ejemplo, tal como se muestra en la figura 1, el almacenamiento 108 incluye un almacén de fragmentos 118. La interfaz de almacén de fragmentos 116 puede almacenar los fragmentos de datos recibidos de los fragmentos de datos 124 en el almacén de fragmentos 118 como los fragmentos de datos 128.

La API de acceso de datos 114 proporciona una interfaz para que las aplicaciones soliciten datos del sistema de almacenamiento 102. Por ejemplo, tal como se muestra en la figura 1, la API de acceso de datos 114 puede recibir una solicitud de flujo de datos 120. La API de acceso de datos 114 se puede configurar de cualquier forma adecuada, tal como sería conocido por los expertos en la materia o materias relevantes. La API de acceso de datos 114 puede emitir la solicitud de flujo de datos 120 para que sea recibida por la interfaz de almacén de fragmentos 116. La interfaz de almacén de fragmentos 116 puede solicitar los fragmentos de datos a partir del almacenamiento 108 (por ejemplo, a partir del almacén de fragmentos 118) que se corresponden con el flujo de datos solicitado de la solicitud de flujo de datos 120. La interfaz de almacén de fragmentos 116 puede recibir los fragmentos de datos solicitados a partir del almacenamiento 108 como los fragmentos de datos 130, y puede proporcionar un flujo de datos que incluye los fragmentos de datos 130 a la API de acceso de datos 114. La API de acceso de datos 114 puede proporcionar el flujo de datos (por ejemplo, un archivo o archivos vueltos a ensamblar) a la aplicación solicitante como una respuesta de flujo de datos 122.

Además, el módulo de mantenimiento 106 se puede encontrar presente para realizar uno o más tipos de labores de mantenimiento con respecto a los fragmentos de datos que se almacenan en el almacén de fragmentos 118. Por ejemplo, el módulo de mantenimiento 106 puede incluir un módulo de desfragmentación para realizar la desfragmentación de fragmentos de datos que se almacenan en el almacenamiento 108. Por ejemplo, el módulo de desfragmentación se puede configurar para eliminar los espacios vacíos en el almacenamiento 108 (por ejemplo, realizar una compactación), para mover fragmentos de datos relacionados a una secuencia y/o para realizar otras tareas relacionadas. En otro ejemplo, el módulo de mantenimiento 106 puede incluir un módulo de recolección de elementos no usados para realizar una recolección de elementos no usados de fragmentos de datos que se

almacenar en el almacenamiento 108. Por ejemplo, el módulo de recolección de elementos no usados se puede configurar para eliminar fragmentos de datos sin usar en el almacenamiento 108. En unas realizaciones adicionales, el módulo de mantenimiento 106 puede realizar unas tareas de mantenimiento adicionales o alternativas con respecto al almacenamiento 108.

5 Tal como se muestra en la figura 1, la API de mantenimiento de fragmentos 112 proporciona una interfaz para que el módulo de mantenimiento 106 interaccione con el sistema de almacenamiento 102. El módulo de mantenimiento 106 puede generar una tarea de mantenimiento 126 (por ejemplo, una instrucción de desfragmentación, una instrucción de compactación, una instrucción de eliminación de fragmento de datos, etc.) que es recibida por la API de mantenimiento de fragmentos 112. La API de mantenimiento de fragmentos 112 se puede configurar de cualquier
10 forma adecuada, tal como sería conocido por los expertos en la materia o materias relevantes. La API de mantenimiento de fragmentos 112 puede proporcionar la tarea de mantenimiento 126 a la interfaz de almacén de fragmentos 116. La interfaz de almacén de fragmentos 116 puede posibilitar que la tarea de mantenimiento 126 se realice sobre fragmentos de datos que se almacenan en el almacenamiento 108.

15 El sistema de almacenamiento 102 se puede implementar en cualquier forma adecuada, incluyendo en forma de uno o más ordenadores/dispositivos informáticos, etc. El almacenamiento 108 puede incluir uno o más de cualquier tipo de mecanismo de almacenamiento, incluyendo un disco magnético (por ejemplo, en una unidad de disco duro), un disco óptico (por ejemplo, en una unidad de disco óptico), una cinta magnética (por ejemplo, en una unidad de cinta), y/o cualquier otro tipo adecuado de medio de almacenamiento.

20 Obsérvese que el sistema de deduplicación de datos 100 es un ejemplo de un entorno en el que se pueden implementar realizaciones de la presente invención. El sistema de deduplicación de datos 100 se proporciona por razones de ilustración, y no tiene por objeto ser limitante. Se pueden incorporar realizaciones en tipos y configuraciones adicionales de los sistemas de deduplicación de datos.

B. Realizaciones de almacén de fragmentos que posibilitan una Localidad de fragmento de datos a modo de ejemplo

25 El almacén de fragmentos 118 de la figura 1 puede almacenar flujos de datos en forma de fragmentos de datos de cualquier forma. Por ejemplo, el almacén de fragmentos 118 puede almacenar metadatos de flujo optimizado en forma de asignaciones, bases de datos u otras formas, que indican los fragmentos de datos que se incluyen en los flujos de datos. El almacén de fragmentos 118 puede almacenar adicionalmente los fragmentos de datos a los que se hace referencia. En una realización, el almacén de fragmentos 118 no almacena copias duplicadas de fragmentos de datos, de acuerdo con las técnicas de deduplicación de datos.

30 Por ejemplo, la figura 2 muestra un diagrama de bloques del almacén de fragmentos 118, de acuerdo con una realización a modo de ejemplo. Tal como se muestra en la figura 2, el almacén de fragmentos 118 incluye un contenedor de flujo 202 y un contenedor de fragmentos 204. El contenedor de flujo 202 incluye los metadatos de flujo optimizado 206 para uno o más flujos de datos, y el contenedor de fragmentos 204 incluye una pluralidad de fragmentos de datos 208. Un fragmento de datos 208 es un segmento de datos al que hacen referencia uno o más
35 flujos de datos (por ejemplo, el flujo de datos 132 de la figura 1). Los metadatos de flujo optimizado 206 son una estructura de datos que describe la asignación entre la estructura de flujo de datos original y la estructura de fragmentos de datos optimizada. Los metadatos de flujo optimizado 206 contienen una información de ubicación de fragmentos de datos, o bien directamente o bien a través de una capa de direccionamiento indirecto, de tal modo que los fragmentos de datos a los que se hace referencia se pueden localizar y ensamblar en una vista de flujo de
40 archivo. Los metadatos de flujo optimizado 206 pueden tener varias formas en unas realizaciones, incluyendo tener forma de asignaciones de flujo (indicando cada asignación de flujo unas ubicaciones de fragmentos de datos para un flujo de datos correspondiente), una base de datos o una tabla global (que indica unas ubicaciones de fragmentos de datos para todos los flujos de datos) u otra forma. En el ejemplo de la figura 2, los fragmentos de datos 208 y los metadatos de flujo optimizado 206 se almacenan en el contenedor de flujo 202 y el contenedor de fragmentos 204,
45 de forma respectiva, los cuales pueden ser archivos en un sistema de archivos. En una realización, el almacén de fragmentos 118 puede almacenar todos los datos en forma de fragmentos, de tal modo que los metadatos de flujo optimizado 206 se almacenan como fragmentos de datos (por ejemplo, cada asignación de flujo se almacena como un fragmento de datos) que contienen unos metadatos internos (metadatos de flujo de datos) para describir la asignación de flujos de archivo con respecto a los fragmentos de datos 208, direcciones de fragmentos de datos y valores de troceo. Como alternativa, los metadatos de flujo optimizado 206 se pueden almacenar en otras formas
50 (por ejemplo, una tabla o base de datos central, etc.).

El contenedor de flujo 202 y el contenedor de fragmentos 204 se pueden configurar de varias formas, en unas realizaciones. Por ejemplo, la figura 3 muestra un diagrama de bloques de un almacén de fragmentos 300, de acuerdo con una realización a modo de ejemplo. El almacén de fragmentos 300 es un ejemplo del almacén de
55 fragmentos 118 de la figura 2. Tal como se muestra en la figura 3, el almacén de fragmentos 300 incluye un contenedor de almacenamiento 302 y un contenedor de fragmentos 304. El contenedor de almacenamiento 302 es un ejemplo del contenedor de almacenamiento 202 de la figura 2, y el contenedor de fragmentos 304 es un ejemplo del contenedor de fragmentos 204 de la figura 2. En la realización de la figura 3, el contenedor de almacenamiento 302 incluye un encabezado de archivo 306, una tabla de redirección 308, y una pluralidad de asignaciones de flujo
60 310. Las asignaciones de flujo 310 son unos ejemplos de los metadatos de flujo optimizado 206 que se proporcionan

para facilitar la ilustración. Por razones de ilustración, los metadatos de flujo optimizado 206 se pueden describir con frecuencia en el presente documento en términos de las asignaciones de flujo, pero se tiene por objeto que se entienda que, en otras realizaciones, los metadatos de flujo optimizado 206 se pueden almacenar, como alternativa, como una base de datos, una tabla global, etc.

5 La primera y la segunda asignaciones de flujo 310a y 310b se muestran en la figura 3 por razones de ilustración, pero en unas realizaciones, se puede incluir cualquier número de asignaciones de flujo 310 en el contenedor de flujo 302, incluyendo cientos, miles e incluso números mayores de asignaciones de flujo 310. El contenedor de fragmentos 304 incluye un encabezado de archivo 318, una tabla de redirección 320, y una pluralidad de fragmentos de datos 322. El primer y el segundo fragmentos de datos 322a y 322b se muestran en la figura 3 por razones de
10 ilustración, pero en unas realizaciones, se puede incluir cualquier número de fragmentos de datos 322 en el contenedor de fragmentos 304, incluyendo cientos, miles e incluso números mayores de fragmentos de datos 322. Estos rasgos distintivos de la figura 3 se describen tal como sigue.

El encabezado de archivo 306 es un encabezado de archivo para el contenedor de flujo 302 en una realización en la que el contenedor de flujo 302 se almacena como un archivo. El encabezado de archivo 306 puede incluir una
15 información que está asociada con el contenedor de flujo 302, incluyendo un identificador de contenedor de flujo (por ejemplo, un número de identificación de contenedor de flujo), etc.

Opcionalmente, la tabla de redirección 308 se encuentra presente en el contenedor de flujo 302. Cuando se encuentra presente, la tabla de redirección 308 puede almacenar una información con respecto a cambios en la ubicación en el contenedor de flujo 302 de cualquiera de las asignaciones de flujo 310. Por ejemplo, la primera
20 asignación de flujo 310a se puede eliminar del contenedor de flujo 302, y la segunda asignación de flujo 310b puede ser movida a la ubicación de la primera asignación de flujo 310a (por ejemplo, debido a una rutina de desfragmentación o de compactación). Posterior al movimiento, se puede acceder al contenedor de flujo 302 mediante una aplicación para recuperar la segunda asignación de flujo 310b. No obstante, la aplicación puede seguir usando la ubicación anterior de la segunda asignación de flujo 310b. La tabla de redirección 308 puede incluir una
25 asignación para la segunda asignación de flujo 310b que indica la ubicación actual de la segunda asignación de flujo 310b. En ese sentido, la aplicación puede acceder a la tabla de redirección 308 para determinar la ubicación actual de la segunda asignación de flujo 310b, y se puede habilitar de ese modo para recuperar la segunda asignación de flujo 310b a partir de su nueva ubicación.

Las asignaciones de flujo 310 son unos ejemplos de los metadatos de flujo optimizado 206 de la figura 2. Cada una
30 de las asignaciones de flujo 310 se usa para definir las secuencias de los fragmentos de datos 322 que constituyen un flujo de datos particular. Tal como se muestra en la figura 3, Cada una de las asignaciones de flujo 310 puede incluir un encabezado de flujo 312, unos metadatos 314 y unos valores de troceo 316. Por ejemplo, la primera asignación de flujo 310a se muestra incluyendo el encabezado de flujo 312a, los metadatos 314a y los valores de troceo 316a, y la segunda asignación de flujo 310b se muestra incluyendo el encabezado de flujo 312b, los
35 metadatos 314b y los valores de troceo 316b. Cada encabezado de flujo 312 incluye una información que está asociada con la asignación de flujo 310 correspondiente, tal como un identificador de asignación de flujo (por ejemplo, un número de identificación de asignación de flujo), etc. Cada uno de los metadatos 314 incluye una información que describe los fragmentos de datos 322 que constituyen el flujo de datos que es definido por la asignación de flujo 310 correspondiente. Opcionalmente, se encuentran presentes los valores de troceo 316. Los
40 valores de troceo 316 son valores de troceo para los fragmentos de datos 322 que constituyen el flujo de datos que es definido por la asignación de flujo 310 correspondiente. Los valores de troceo 316 se pueden almacenar en las asignaciones de flujo 310 con el fin de proporcionar un acceso eficiente a un vector de troceo de los fragmentos de datos que constituyen el flujo de datos correspondiente. Por ejemplo, esto puede ser útil para escenarios de transferencia de datos por cable en los que se desea un acceso rápido a una lista completa de valores de troceo de
45 flujo de datos (valores de troceo para la totalidad de los fragmentos de archivo optimizados).

Los metadatos 314 son unos metadatos en función de los fragmentos de datos, o unos metadatos específicos de fragmentos de datos, que se pueden incluir en la asignación de flujo 310 para cada fragmento de datos 322 al que se hace referencia. Se pueden incluir varios tipos de información en los metadatos 314. Por ejemplo, en una
50 realización, los metadatos 314 para un fragmento de datos 322 pueden incluir uno o más de un desplazamiento de flujo de datos, un identificador de fragmento de datos y un indicador de localidad. El desplazamiento de flujo de datos indica una ubicación para el fragmento de datos 322 asociado en el flujo de datos que es definido por la asignación de flujo 310 particular (por ejemplo, un número de bytes desde el inicio del flujo de datos, o desde otro punto de referencia en el flujo de datos, en el cual comienza el fragmento de datos 322 asociado). El identificador de fragmento de datos, que también se conoce como un ID de fragmento o "localizador de fragmento fiable", es una
55 referencia o puntero al fragmento de datos 322 correspondiente en el contenedor de fragmentos 304. El indicador de localidad es una información que representa un orden de inserción de fragmentos en el contenedor de fragmentos 304, posibilitando que se haga una determinación de a qué fragmentos de datos 322 se puede hacer referencia mediante una asignación de flujo 310 común. En otras realizaciones, los metadatos 314 pueden incluir una información adicional y/o alternativa para cada fragmento de datos 322 al que se hace referencia.

60 Con referencia al contenedor de fragmentos 304 de la figura 3, el encabezado de archivo 318 es un encabezado de archivo para el contenedor de fragmentos 302 en una realización en la que el contenedor de fragmentos 304 se

almacena como un archivo. El encabezado de archivo 318 puede incluir una información que está asociada con el contenedor de fragmentos 304, incluyendo un identificador de contenedor de fragmentos (por ejemplo, un número de identificación de contenedor de fragmentos), un indicador de generación de contenedor de fragmentos que indica un número de revisión del contenedor de fragmentos 304, etc.

5 Opcionalmente, la tabla de redirección 320 se encuentra presente en el contenedor de fragmentos 304. Cuando se encuentra presente, la tabla de redirección 320 puede almacenar una información con respecto a cambios en la ubicación en el contenedor de fragmentos 304 de cualquiera de los fragmentos de datos 322, de una forma similar a cómo la tabla de redirección 308 del contenedor de flujo 302 maneja cambios en la ubicación de las asignaciones de flujo 310.

10 Los fragmentos de datos 322 son unos ejemplos de los fragmentos de datos 208 de la figura 2. Tal como se muestra en la figura 3, cada uno de los fragmentos de datos 322 incluye un encabezado de fragmento 324 y unos datos de fragmento 326. Por ejemplo, el primer fragmento de datos 322a incluye el encabezado de fragmento 324a y los datos de fragmento 326a, y el segundo fragmento de datos 322b incluye el encabezado de fragmento 324b y los datos de fragmento 326b. Cada encabezado de flujo 312 incluye una información que está asociada con el
15 fragmento de datos 322 correspondiente, tal como un identificador de fragmento de datos, etc. Cada uno de los datos de fragmento 326 incluye los datos correspondientes, los cuales se pueden encontrar en forma comprimida o no comprimida.

Las asignaciones de flujo 310 y los fragmentos de datos 322 se almacenan en el contenedor de flujo 302 y el contenedor de fragmentos 304, de forma respectiva, para posibilitar una deduplicación de datos. Por ejemplo, la
20 interfaz de almacén de fragmentos 116 de la figura 1 puede recibir los fragmentos de datos 124 que están asociados con los flujos de datos 132 y puede almacenar los fragmentos de datos en el almacén de fragmentos 300 de la figura 3. Por ejemplo, para un flujo de datos 132 particular, la interfaz de almacén de fragmentos 116 puede generar una asignación de flujo que se almacena en el contenedor de flujo 302 como una asignación de flujo 310 que hace referencia a uno o más fragmentos de datos 322 que se almacenan en el contenedor de fragmentos 304 mediante la
25 interfaz de almacén de fragmentos 116.

Por ejemplo, la figura 3 indica algunos fragmentos de datos 322 a los que hacen referencia las asignaciones de flujo 310, de acuerdo con una realización a modo de ejemplo. Tal como se muestra en la figura 3, la primera asignación de flujo 310a incluye los metadatos 314a que incluyen referencias al primer y el segundo fragmentos de datos 322a y 322b en el contenedor de fragmentos 304. Por lo tanto, el primer y el segundo fragmentos de datos 322a y 322b se
30 incluyen en el flujo de datos de origen que está asociado con la primera asignación de flujo 310a. Por ejemplo, los metadatos 314a pueden incluir un valor del desplazamiento de flujo de datos 402 para el primer fragmento de datos 322a que indica una ubicación del primer fragmento de datos 322a en el flujo de datos de origen que es definido por la primera asignación de flujo 310a, un identificador de fragmento de datos 404 para el primer fragmento de datos 322a en el contenedor de fragmentos 304 (por ejemplo, el identificador de fragmento de datos para el primer
35 fragmento de datos 322a que se almacena en el encabezado de fragmento 324a), y un indicador de localidad 406 para el primer fragmento de datos 322a. Además, los metadatos 314a pueden incluir un valor del desplazamiento de flujo de datos 402 para el segundo fragmento de datos 322b que indica una ubicación del segundo fragmento de datos 322b en el flujo de datos de origen, un identificador de fragmento de datos 404 para el segundo fragmento de datos 322b en el contenedor de fragmentos 304 (por ejemplo, el identificador de fragmento de datos para el segundo
40 fragmento de datos 322b que se almacena en el encabezado de fragmento 324b), y un indicador de localidad 406 para el segundo fragmento de datos 322b. En una realización, el primer y el segundo fragmentos de datos 322a y 322b pueden tener un mismo valor para sus indicadores de localidad que se genera para corresponderse con el flujo de datos de origen que es definido por la primera asignación de flujo 310a, y que indica que el primer y el segundo fragmentos de datos 322a y 322b se almacenan de forma contigua (de forma adyacente) en el contenedor de
45 fragmentos 304.

Además, la segunda asignación de flujo 310b incluye los metadatos 314b que incluyen referencias al segundo fragmento de datos 322b en el contenedor de fragmentos 304. Por ejemplo, los metadatos 314b pueden incluir un valor del desplazamiento de flujo de datos 402 para el segundo fragmento de datos 322b que indica una ubicación del segundo fragmento de datos 322b en el flujo de datos de origen que es definido por la segunda asignación de
50 flujo 310b, un identificador de fragmento de datos 404 para el segundo fragmento de datos 322b en el contenedor de fragmentos 304 (por ejemplo, el identificador de fragmento de datos para el segundo fragmento de datos 322b que se almacena en el encabezado de fragmento 324b), y un indicador de localidad 406 para el segundo fragmento de datos 322b. El indicador de localidad 406 en los metadatos 314b para el segundo fragmento de datos 322b tiene el mismo valor que los indicadores de localidad que se generan para el primer y el segundo fragmentos de datos 322a y 322b debido a que el segundo fragmento de datos 322b se almacenó originalmente en el contenedor de fragmentos 304 para la primera asignación de flujo 310a. A cualesquiera fragmentos de datos 322 adicionales (que no se muestran en la figura 3) que estuvieran recién almacenados en el contenedor de fragmentos 304 cuando el flujo de datos de origen que es definido por la segunda asignación de flujo 310b se almacenó en el almacén de fragmentos 300 se les asigna un nuevo valor para el indicador de localidad 406.

60 La interfaz de almacén de fragmentos 116 de la figura 1 se puede configurar de varias formas para almacenar flujos de datos en el almacén de fragmentos 300 de la figura 3. Por ejemplo, la figura 4 muestra un diagrama de bloques

del sistema de almacén de flujo de datos 400, de acuerdo con una realización a modo de ejemplo. Tal como se muestra en la figura 4, el sistema de almacén de flujo de datos 400 incluye una unidad de análisis sintáctico de flujo de datos 402, una interfaz de almacén de fragmentos 116, un contenedor de flujo 302 y un contenedor de fragmentos 304. En una realización, la unidad de análisis sintáctico de flujo de datos 402 se puede incluir en el módulo de deduplicación de datos 104 de la figura 1. En la realización de la figura 4, la interfaz de almacén de fragmentos 116 incluye un gestor de almacenamiento de fragmentos de datos 404, un generador de metadatos 406 y un generador de asignaciones de flujo 408. Estos rasgos distintivos de la figura 4 se describen tal como sigue con respecto a la figura 5. La figura 5 muestra un diagrama de flujo 500 para almacenar un flujo de datos, de acuerdo con una realización a modo de ejemplo. En una realización, el sistema 400 de la figura 4 puede funcionar de acuerdo con el diagrama de flujo 500. Realizaciones estructurales y operativas adicionales resultarán evidentes a los expertos en la materia o materias relevantes basándose en el análisis con respecto al diagrama de flujo 500. El diagrama de flujo 500 y el sistema 400 se describen tal como sigue.

El diagrama de flujo 500 comienza con la etapa 502. En la etapa 502, un flujo de datos se somete a análisis sintáctico en fragmentos de datos. Por ejemplo, tal como se muestra en la figura 4, la unidad de análisis sintáctico de flujo de datos 402 puede recibir un flujo de datos 410. El flujo de datos 410 puede incluir uno o más archivos y/o porciones de archivos, de forma similar al flujo de datos 132 de la figura 1. La unidad de análisis sintáctico de flujo de datos 402 está configurada para analizar sintácticamente el flujo de datos 410 en una secuencia de fragmentos de datos, que se indica como la secuencia de fragmentos de datos 412. Por ejemplo, en una realización, la secuencia de fragmentos de datos 412 puede incluir la secuencia de fragmentos de datos en el orden en el que están ubicados los fragmentos de datos en el flujo de datos 410. Los fragmentos de datos de la secuencia de fragmentos de datos 412 pueden tener el mismo tamaño o pueden tener diferentes tamaños.

En la etapa 504, se determina si cualquiera de los fragmentos de datos son duplicados de fragmentos de datos que se almacenan en un contenedor de fragmentos. Por ejemplo, tal como se muestra en la figura 4, el gestor de almacenamiento de fragmentos de datos 404 recibe la secuencia de fragmentos de datos 412. El gestor de almacenamiento de fragmentos de datos 404 está configurado para determinar si cualquiera de los fragmentos de datos de la secuencia de fragmentos de datos 412 ya están almacenados en el contenedor de fragmentos 304 y, por lo tanto, son duplicados. Por ejemplo, en una realización, tal como se muestra en la figura 4, el gestor de almacenamiento de fragmentos de datos 404 puede recibir la información de fragmento de datos 426 a partir del contenedor de fragmentos 304, el cual puede incluir valores de troceo para cada fragmento de datos 322 que se almacena en el contenedor de fragmentos 304. En otra realización, el gestor de almacenamiento de fragmentos de datos 404 puede recibir los valores de troceo 316 (la figura 3) a partir del contenedor de flujo 302, los cuales son valores de troceo para los fragmentos de datos 322 que se almacenan en el contenedor de fragmentos 304. El gestor de almacenamiento de fragmentos de datos 404 puede generar un valor de troceo para cada fragmento de datos de la secuencia de fragmentos de datos 412, y puede comparar los valores de troceo generados con los valores de troceo que se reciben en la información de fragmento de datos 426 (o a partir del contenedor de flujo 302) para determinar qué fragmentos de datos de la secuencia de fragmentos de datos 412 ya están almacenados en el contenedor de fragmentos 304. En unas realizaciones adicionales, el gestor de almacenamiento de fragmentos de datos 404 puede determinar qué fragmentos de datos de la secuencia de fragmentos de datos 412 ya están almacenados en el contenedor de fragmentos 304 de otras formas, tal como sería conocido por los expertos en la materia o materias relevantes.

Tal como se muestra en la figura 4, el gestor de almacenamiento de fragmentos de datos 404 genera una indicación de fragmentos almacenados 416, la cual indica qué fragmentos de datos de la secuencia de fragmentos de datos 412 ya están almacenados en el contenedor de fragmentos 304.

Haciendo referencia de nuevo a la figura 5, en la etapa 506, los fragmentos de datos que se determina que no son duplicados se almacenan en el contenedor de fragmentos en una disposición contigua y en una secuencia que es la misma que en el flujo de datos. Por ejemplo, en una realización, el gestor de almacenamiento de fragmentos de datos 404 se puede configurar para almacenar los fragmentos de datos de la secuencia de fragmentos de datos 412 que no se determinó que estaban almacenados en el contenedor de fragmentos 304. Por ejemplo, en una realización, el gestor de almacenamiento de fragmentos de datos 404 puede generar un encabezado de fragmento 324 (por ejemplo, un identificador de fragmento de datos) para cada nuevo fragmento de datos, y almacenar cada nuevo fragmento de datos como un fragmento de datos 322 con el encabezado de fragmento 324 y los datos de fragmento 326. Además, en una realización, el gestor de almacenamiento de fragmentos de datos 404 está configurado para almacenar los fragmentos de datos nuevos en una disposición contigua en el contenedor de fragmentos 304, en un orden que es el mismo que en el flujo de datos de origen (por ejemplo, en el orden que se recibe en la secuencia de fragmentos de datos 412).

En la etapa 508, se generan metadatos para cada uno de los fragmentos de datos que se determina que no son un duplicado, incluyendo los metadatos para un fragmento de datos un desplazamiento de flujo de datos, un puntero a una ubicación en el contenedor de fragmentos, y un indicador de localidad. Por ejemplo, tal como se muestra en la figura 4, el generador de metadatos 406 puede recibir la secuencia de fragmentos de datos 412 y la indicación de fragmentos almacenados 416. En una realización, el generador de metadatos 406 se puede configurar para generar metadatos (por ejemplo, los metadatos 314 de la figura 3). El generador de metadatos 406 puede generar metadatos para cada fragmento de datos de la secuencia de fragmentos de datos 412, incluyendo el desplazamiento de flujo de

datos 402, el identificador de fragmento de datos 404 y el indicador de localidad 406. Para los fragmentos de datos que se determina que ya están almacenados en el contenedor de fragmentos 304 (en la etapa 504), el identificador de fragmento de datos 404 está configurado para apuntar al fragmento de datos ya almacenado. Para los fragmentos de datos recién almacenados en el contenedor de fragmentos 304 en la etapa 508, el identificador de fragmento de datos 404 está configurado para apuntar al fragmento de datos recién almacenado.

Haciendo referencia de nuevo a la figura 5, en la etapa 510, se genera una asignación de flujo para el flujo de datos que incluye los metadatos generados. Por ejemplo, tal como se muestra en la figura 4, el generador de asignaciones de flujo 408, recibe los metadatos de fragmentos de datos 420 para cada fragmento de datos que se recibe en la secuencia de fragmentos de datos 412 para un flujo de datos particular. El generador de asignaciones de flujo 408 genera una asignación de flujo 424 que está asociada con el flujo de datos que incluye los metadatos de fragmentos de datos 420 para cada fragmento de datos recibido. Además, el generador de asignaciones de flujo 408 puede generar un encabezado de flujo 312 para la asignación de flujo 424, y puede incluir los valores de troceo 316 para cada fragmento de datos recibido en la asignación de flujo 424.

En la etapa 512, la asignación de flujo se almacena en un contenedor de flujo. Por ejemplo, tal como se muestra en la figura 4, el generador de asignaciones de flujo 408 puede almacenar (o dar "persistencia" a) la asignación de flujo 424 en el contenedor de flujo 302 (por ejemplo, como una asignación de flujo 310). Obsérvese que, en una realización alternativa, en lugar de generar y almacenar una asignación de flujo para un flujo de datos, se puede hacer una entrada en una base de datos o una tabla global para el flujo de datos que incluye unos metadatos que apuntan a o que indican una ubicación de los fragmentos de datos a los que hace referencia el flujo de datos.

La figura 6 muestra un diagrama de bloques que ilustra un ejemplo del almacenamiento de flujos de datos en un almacén de datos, de acuerdo con una realización. La figura 6 se proporciona por razones de ilustración, y no tiene por objeto ser limitante. En el ejemplo de la figura 6, se almacena un primer flujo de datos 602a en un almacén de datos, seguido por el almacenamiento de un segundo flujo de datos 602b en el almacén de datos. Se muestra un vínculo de flujo 608a (que también se conoce como "puntero de flujo" o "código auxiliar de metadatos de flujo") para el primer flujo de datos 602a, y se muestra un vínculo de flujo 608b para el segundo flujo de datos 602b. Cada vínculo de flujo 608 vincula un flujo de datos 602 con los datos correspondientes (por ejemplo, la asignación de flujo 604 u otros metadatos de flujo optimizado) en el almacén de fragmentos que posibilita que se vuelva a ensamblar el flujo de datos 602. Tal como se muestra en la figura 6, el primer flujo de datos 602a incluye cuatro fragmentos de datos 614a - 614d. Se puede generar una asignación de flujo 604a para el primer flujo de datos 602a, y los cuatro fragmentos de datos 614a - 614d se pueden almacenar en un contenedor de fragmentos 606, tal como se ha descrito en lo que antecede. La asignación de flujo 604a incluye punteros (que se representan mediante flechas en la figura 6) a cada uno de los fragmentos de datos 614a - 614d. Los fragmentos de datos 614a - 614d se pueden categorizar en un único conjunto de todos los fragmentos de datos únicos y nuevos para el contenedor de fragmentos 606. En ese sentido, los fragmentos de datos 614a - 614d se pueden almacenar en el contenedor de fragmentos 606 en una disposición contigua, en un orden que es el mismo que en el flujo de datos 602a. Por ejemplo, los fragmentos de datos 614a - 614d pueden ser los primeros cuatro fragmentos de datos que se almacenan en el contenedor de fragmentos 606, o si uno o más fragmentos de datos ya están almacenados en el contenedor de fragmentos 606, los fragmentos de datos 614a - 614d se pueden almacenar en el contenedor de fragmentos 606 inmediatamente después de los fragmentos de datos ya almacenados. A cada uno de los fragmentos de datos 614a - 614d se les asigna el mismo valor de indicador de localidad en la asignación de flujo 604a, el valor de indicador de localidad que se selecciona para el primer flujo de datos 602a.

El segundo flujo de datos 602b incluye cuatro fragmentos de datos 614b, 614c, 614e y 614f. Se puede generar una asignación de flujo 604b para el segundo flujo de datos 602b. Los fragmentos de datos 614b, 614c, 614e y 614f se pueden categorizar en dos conjuntos de fragmentos de datos de acuerdo con la etapa 504 del diagrama de flujo 500: un primer conjunto que incluye los fragmentos 614b y 614c, los cuales ya tienen copias residuando en el contenedor de fragmentos 606 (debido a la secuencia de fragmentos del primer flujo de datos 602a), y un segundo conjunto que incluye los fragmentos 614e y 614f, los cuales son unos fragmentos de datos únicos y nuevos (que no tienen copias ya almacenadas en el contenedor de fragmentos 606). Debido a que los fragmentos de datos 614b y 614c ya están almacenados en el contenedor de fragmentos 606, la asignación de flujo 604b incluye punteros (valores para el identificador de fragmento de datos 404) a los fragmentos de datos 614b y 614c que ya están almacenados en el contenedor de fragmentos 606. Por lo tanto, los fragmentos de datos 614b y 614c se pueden almacenar como punteros a fragmentos de datos existentes en el contenedor de fragmentos 606 sin almacenar datos de fragmento de los fragmentos de datos 614b y 614c. Debido a que los fragmentos de datos 614e y 614f no están almacenados aún en el contenedor de fragmentos 606, los fragmentos de datos 614e y 614f se pueden almacenar en el contenedor de fragmentos 606, tal como se ha descrito en lo que antecede. Por ejemplo, debido a que los fragmentos de datos 614e y 614f son unos fragmentos de datos únicos y nuevos para el contenedor de fragmentos 606, los fragmentos 614e y 614f se pueden almacenar en el contenedor de fragmentos 606 en una disposición contigua, en un orden que es el mismo que en el flujo de datos 602b, después del último fragmento de datos almacenado que está actualmente almacenado en el contenedor de fragmentos 606 (por ejemplo, el fragmento de datos 614d). La asignación de flujo 604b incluye de un primer a un cuarto identificadores de fragmento de datos 612a - 612d, los cuales apuntan a los fragmentos de datos 614b, 614c, 614e y 614f que se almacenan en el contenedor de fragmentos 606, de forma respectiva. En la asignación de flujo 604b, a los fragmentos de datos 614b y 614c se les asigna el valor de indicador de localidad que está asociado con el primer flujo de datos 602a y a los

fragmentos de datos 614e y 614f se les asigna el valor de indicador de localidad que se selecciona para el segundo flujo de datos 602b.

Obsérvese que cualquier número de flujos de datos 602 adicionales se pueden almacenar de una forma similar siguiendo los flujos de datos 602a y 602b. Además, obsérvese que, en el ejemplo de la figura 6, a cada uno de los fragmentos de datos de la segunda asignación de flujo 604b se le asignó uno de dos valores de indicador de localidad - o bien el nuevo valor de indicador de localidad que se selecciona para la segunda asignación de flujo 604b, o bien el valor de indicador de localidad que está asociado con los fragmentos de datos de la primera asignación de flujo 604a. En unas realizaciones, a los fragmentos de datos de una asignación de flujo particular se les puede asignar uno de cualquier número de valores de indicador de localidad, dependiendo del número de indicadores de localidad diferentes que están asociados con fragmentos de datos de la asignación de flujo que ya se encuentran presentes en el contenedor de fragmentos. Por ejemplo, tal como se ha descrito en lo que antecede, a los fragmentos de datos nuevos para un contenedor de fragmentos se les puede asignar el nuevo valor de indicador de localidad que se selecciona para el flujo de datos particular que está asociado con la asignación de flujo. Además, a cualquier número de fragmentos de datos a los que hace referencia la asignación de flujo que ya se encuentran presentes en el contenedor de fragmentos se les asignan los valores de indicador de localidad correspondientes de los fragmentos de datos que ya se encuentran presentes en el contenedor de fragmentos. Esto puede querer decir que a cualquier número de uno o más conjuntos de fragmentos de datos del flujo de datos se les puede asignar unos valores de indicador de localidad correspondientes, de tal modo que a los fragmentos de datos del flujo de datos se les puede asignar unos indicadores de localidad que se seleccionan de entre dos, tres, o incluso más valores de indicador de localidad diferentes.

En ese sentido, los indicadores de localidad de los metadatos de asignación de flujo posibilitan que se averigüe la localidad de fragmentos de datos en flujos de datos. Esto es debido a que los fragmentos de datos duplicados tienden a tener lugar en grupos. Cuando un nuevo flujo de datos contiene un fragmento de datos ya conocido (que ya está almacenado en el contenedor de fragmentos), hay una probabilidad razonable de que el siguiente fragmento de datos en el nuevo flujo de datos sea también un fragmento de datos duplicado (que ya está almacenado en el contenedor de fragmentos). Debido a que los fragmentos de datos nuevos y originales se almacenan en el contenedor de fragmentos adyacentes entre sí de acuerdo con el indicador de localidad, es más probable que los fragmentos de datos que ya se encuentran presentes a los que hace referencia el nuevo flujo de datos se almacenen también de forma contigua en el contenedor de fragmentos. Esto ayuda a mejorar el rendimiento de la lectura de flujos de datos optimizados a partir de un almacén de fragmentos. Por ejemplo, un módulo de rehidratación que está configurado para volver a ensamblar un flujo de datos basándose en la asignación de flujo y los fragmentos de datos correspondientes puede realizar una lectura anticipada sobre los fragmentos de datos que se almacenan en el contenedor de fragmentos, esperando hallar las siguientes necesidades de fragmentos de datos en la memoria intermedia leída de forma anticipada. Además, las tareas de mantenimiento de almacén de fragmentos como desfragmentación y compactación pueden realizar sus tareas al tiempo que se intenta mantener la localidad original al mantener los fragmentos adyacentes existentes juntos a medida que estos se mueven en torno al contenedor de fragmentos.

Por ejemplo, después de que los flujos de datos se hayan optimizado y almacenado en el almacén de fragmentos 300 en forma de asignaciones de flujo 310 y fragmentos de datos 322, los flujos de datos se pueden leer a partir del almacén de fragmentos 300. La figura 7 muestra un diagrama de bloques de la interfaz de almacén de fragmentos 116 que incluye un módulo de rehidratación 702, de acuerdo con una realización a modo de ejemplo. El módulo de rehidratación 702 está configurado para volver a ensamblar un flujo de datos solicitado (por ejemplo, que se solicita de acuerdo con la solicitud de flujo de datos 120 que se muestra en la figura 1). Por ejemplo, para un flujo de datos que se va a leer a partir del almacén de fragmentos 300 en respuesta a una solicitud de flujo de datos 120 (la figura 1), el módulo de rehidratación 702 determina y recibe la asignación de flujo 310 a la que hace referencia el archivo optimizado de la solicitud de flujo de datos 120 a partir del almacén de fragmentos 300 (por ejemplo, en la ubicación de repetición de análisis sintáctico). Por ejemplo, el módulo de rehidratación 702 puede proporcionar un identificador de asignación de flujo de la solicitud 120 al almacén de fragmentos 300 de la figura 3. El almacén de fragmentos 300 recupera la asignación de flujo 310 correspondiente basándose en el identificador de asignación de flujo (por ejemplo, mediante la exploración de los encabezados de asignación de flujo 312), y el módulo de rehidratación 702 puede regenerar o "rehidratar" el flujo de datos de acuerdo con la asignación de flujo 310 recuperada. La asignación de flujo 310 recuperada incluye punteros (el identificador de fragmento de datos 404 de la figura 4) a cada uno de los fragmentos de datos en el contenedor de fragmentos 304 que se incluyen en el flujo de datos. El módulo de rehidratación 702 usa los punteros para recuperar cada uno de los fragmentos de datos 322. El módulo de rehidratación 702 puede usar los desplazamientos de flujo de datos 402 que se incluyen en la asignación de flujo 310 recuperada (por ejemplo, más una información de longitud de fragmento de datos que se puede incluir en la asignación de flujo 310 recuperada) para disponer los fragmentos de datos 322 recuperados en el orden apropiado para regenerar el flujo de datos, el cual es emitido por el módulo de rehidratación 702 como el flujo de datos 704.

A través del uso de los indicadores de localidad 406, se pueden realizar unas lecturas secuenciales de los fragmentos de datos 322 a partir del contenedor de fragmentos 304. Por ejemplo, cuando el módulo de rehidratación 702 está accediendo a un flujo de archivo en el almacén de fragmentos 300 usando solicitudes secuenciales de E/S (entrada/salida), o cualesquiera solicitudes de E/S que engloben más de un límite de fragmento de datos, la asignación de flujo 310 posibilita un acceso rápido a los fragmentos de datos. Esto es debido a que, en el instante en

el que el almacén de fragmentos 300 crea las asignaciones de flujo 310, se almacenan fragmentos de datos nuevos en el contenedor de fragmentos 304 de forma contigua en el orden de asignación de flujo. En ese sentido, durante un acceso de datos secuencial por el módulo de rehidratación 702, es probable que los fragmentos de datos que pertenecen al mismo flujo de datos se almacenen de forma contigua, se puede acceder a tales fragmentos de datos contiguos y estos se pueden leer con una única "búsqueda" de acceso de datos (por ejemplo, movimientos hacia delante o hacia atrás a través de un contenedor de fragmentos para hallar un fragmento de datos almacenado siguiente que leer), y la fragmentación se reduce a fragmentos de datos no únicos (los fragmentos de datos a los que hace referencia una asignación de flujo que ya se encontraban presentes en el contenedor de fragmentos antes de almacenar el flujo de datos correspondiente). Las búsquedas de acceso de datos durante un acceso de datos secuencial se limitan al caso en el que se halla que un fragmento de datos o una serie de fragmentos de un flujo de datos ya existen en el almacén de fragmentos. La asignación de flujo 310 proporciona un contenedor de metadatos eficiente para los metadatos de archivos optimizados (por ejemplo, los metadatos 314) que pueden que necesiten otros módulos de un sistema de deduplicación de datos (por ejemplo, una lista de valores de troceo que son usados por un módulo de replicación de archivos). Las asignaciones de flujo 310 son concisas y se pueden almacenar de forma temporal en memoria para un acceso rápido. El almacén de fragmentos 300 puede almacenar en memoria caché unas asignaciones de flujo 310 a las que se accede con frecuencia (para flujos de datos optimizados que son solicitados y rehidratados con frecuencia por el módulo de rehidratación 702) basándose en un algoritmo de LRU (*least recently used*, usado menos recientemente) u otro tipo de algoritmo de almacenamiento en memoria caché.

C. Realizaciones de almacén de fragmentos a modo de ejemplo que posibilitan una localización fiable de fragmentos de datos

Tal como se ha descrito en lo que antecede, los fragmentos de datos pueden ser movidos dentro de un contenedor de fragmentos por varias razones, tal como debido a una técnica de desfragmentación, debido a una técnica de compactación que realiza una recolección de elementos no usados, etc. En esta subsección se describen realizaciones para mantener un seguimiento del movimiento de los fragmentos de datos dentro de un contenedor de fragmentos.

La figura 8 muestra un diagrama de bloques del contenedor de fragmentos 304, de acuerdo con una realización a modo de ejemplo. Tal como se muestra en la figura 8, el contenedor de fragmentos 304 es, en general, similar al contenedor de fragmentos 304 de la figura 3, con la adición de un identificador de contenedor de fragmentos 802 y una indicación de generación de contenedor de fragmentos 804 que se incluyen en el encabezado de archivo 318. El identificador de contenedor de fragmentos 802 es un identificador único (por ejemplo, un número de identificación) que se asigna al contenedor de fragmentos 304 para distinguir el contenedor de fragmentos 304 de otros contenedores de fragmentos que se pueden encontrar presentes en el almacén de fragmentos 300. La indicación de generación de contenedor de fragmentos 804 indica una revisión o generación para el contenedor de fragmentos 304. Por ejemplo, cada vez que uno o más fragmentos de datos 322 son movidos dentro del contenedor de fragmentos 304, la indicación de generación 804 se puede modificar (por ejemplo, se puede incrementar hasta un nivel de la siguiente generación, comenzando por un nivel de generación de inicio tal como 0 u otro valor de inicio).

En una realización, el contenedor de fragmentos 304 puede ser identificado por una combinación del identificador de contenedor de fragmentos 802 y la indicación de generación de contenedor de fragmentos 804 (por ejemplo, puede formar un nombre de archivo del contenedor de fragmentos 304). En una realización, tanto el identificador de contenedor de fragmentos 802 como la indicación de generación de contenedor de fragmentos 804 pueden ser números enteros. El contenedor de fragmentos 304 puede tener un tamaño fijo (o un número fijo de entradas), o puede tener un tamaño variable. Por ejemplo, en una realización a modo de ejemplo, cada archivo de contenedor de fragmentos que define un contenedor de fragmentos 304 se puede dimensionar para almacenar aproximadamente 16.000 de los fragmentos, con un tamaño de fragmento de datos promedio de 64 KB, en el que el tamaño del archivo de contenedor de fragmentos se establece a 1 GB. En otras realizaciones, un archivo de contenedor de fragmentos puede tener un tamaño alternativo.

Se puede hacer referencia a los fragmentos de datos 322 que se almacenan en el contenedor de fragmentos 304 de acuerdo con el identificador de fragmento de datos 404 de los metadatos 400 (la figura 4) de varias formas. En unas realizaciones, el identificador de fragmento de datos 404 puede incluir varios tipos de información para posibilitar tal referencia. Por ejemplo, en una realización, el identificador de fragmento de datos 404 puede incluir uno o más de un identificador de contenedor de fragmentos de datos, un identificador local, un valor de generación de contenedor de fragmentos, y un valor de desplazamiento de fragmento. El identificador de contenedor de fragmentos tiene un valor del identificador de contenedor de fragmentos 802 para el contenedor de fragmentos 304 en el que se almacena el fragmento de datos 322. El identificador local es un identificador (por ejemplo, un valor numérico) que se asigna a un fragmento de datos 322, y es único para el fragmento de datos 322 asignado dentro del contenedor de fragmentos 304 en el que se almacena el fragmento de datos 322 (por ejemplo, es un identificador único en función del contenedor para el fragmento de datos). El valor de generación de contenedor de fragmentos tiene el valor de la indicación de generación de contenedor de fragmentos 804 para el contenedor de fragmentos 304 en el que se almacena el fragmento de datos 322, en el instante en el que se almacena el fragmento de datos 322 en el contenedor de fragmentos 304. El valor de desplazamiento de fragmento es un desplazamiento del fragmento de datos 322 en el contenedor de fragmentos 304 en el instante en el que el fragmento de datos 322 se añade al

contenedor de fragmentos 304.

En una realización, un almacén de fragmentos puede implementar un localizador de fragmento fiable que se puede usar para realizar un seguimiento de fragmentos de datos que se han movido. En contraste con las técnicas convencionales, el localizador de fragmento fiable no usa un índice para asignar identificadores de fragmento de datos a una ubicación física de fragmento. Las técnicas convencionales usan un índice que asigna identificadores de fragmento a la ubicación física de datos de fragmento. La escala de los sistemas de almacenamiento (por ejemplo, cientos de Terabytes o mayor) y un tamaño de fragmento promedio (por ejemplo, 64 KB) hacen que un índice de este tipo sea muy grande. Si un índice de este tipo está completamente cargado en memoria, este consumirá una gran cantidad de los recursos disponibles de memoria y de procesador. Si el índice no se carga en memoria, los accesos de datos se vuelven lentos debido a que es necesario que el índice se pague en memoria.

En una realización, el localizador de fragmento fiable se implementa en forma de una tabla de redirección, tal como la tabla de redirección 320 del contenedor de fragmentos 304 en la figura 3. La tabla de redirección puede almacenar una o más entradas para los fragmentos de datos 322 que han sido movidos en el contenedor de fragmentos 304. Cada entrada identifica un fragmento de datos 322 movido, y tiene un valor de desplazamiento de fragmento de datos que indica la ubicación del fragmento de datos 322 en el contenedor de fragmentos 304 en su nueva ubicación. Se puede hacer referencia a la tabla de redirección durante la rehidratación de un flujo de datos para localizar cualesquiera fragmentos de datos del flujo de datos que se han movido.

Por ejemplo, la figura 9 muestra un diagrama de bloques de una tabla de redirección 900, de acuerdo con una realización a modo de ejemplo. La tabla de redirección 900 se usa para localizar los fragmentos de datos 322 (incluyendo asignaciones de flujo que se almacenan como fragmentos de datos) si los fragmentos de datos 322 son movidos dentro del contenedor de fragmentos 304. Por ejemplo, la tabla de redirección 900 posibilita que los fragmentos de datos 322 sean movidos dentro del contenedor de fragmentos 304 para la recuperación de espacio como parte de un procedimiento de recolección y de compactación de elementos no usados, y que se puedan seguir localizando de forma fiable basándose en los identificadores de fragmento originales de los fragmentos de datos 322. Tal como se muestra en la figura 9, la tabla de redirección 900 incluye una pluralidad de entradas 902, tal como una primera entrada 902a y una segunda entrada 902b. Se puede incluir cualquier número de entradas 902 en la tabla de redirección 900, incluyendo cientos, miles e incluso números mayores de entradas 902. Cada entrada 902 incluye un identificador local 904 y un valor de desplazamiento de fragmento cambiado 906. Por ejemplo, la primera entrada 902a incluye un primer identificador local 904a y un primer valor de desplazamiento de fragmento cambiado 906a, y la segunda entrada 902b incluye un segundo identificador local 904b y un segundo valor de desplazamiento de fragmento cambiado 906b.

El identificador local 904 es el identificador local único que se asigna a un fragmento de datos 322 cuando se almacena originalmente en el contenedor de fragmentos 304. El valor de desplazamiento de fragmento cambiado 906 es el nuevo valor de desplazamiento de fragmento para el fragmento de datos 322 que tiene el identificador local 904 correspondiente que fue movido. En ese sentido, se puede acceder a la tabla de redirección 900 usando un indicador de localidad para un fragmento de datos para determinar un valor de desplazamiento de fragmento cambiado para el fragmento de datos.

Por ejemplo, el identificador local 904a en la figura 9 puede ser el identificador local que se asigna al fragmento de datos 614b. Se puede acceder a la entrada 902a de la tabla de redirección 900 usando el identificador local que se asigna al fragmento de datos 614b para determinar el valor de desplazamiento de fragmento cambiado 906a, el cual indica una nueva ubicación para el fragmento de datos 614b en el contenedor de fragmentos 304.

Obsérvese que la tabla de redirección 900 puede tener cualquier tamaño. Por ejemplo, en una realización, el tamaño de la tabla de redirección 900 puede estar limitado por (un número máximo previamente determinado de fragmentos de datos - un número mínimo previamente determinado de fragmentos de datos que se eliminan para su compactación) \times (un tamaño de una entrada de tabla de redirección). En algunos casos, las reubicaciones de fragmentos de datos pueden ser poco frecuentes. En una realización, después de la determinación de un valor de desplazamiento de fragmento cambiado para un fragmento de datos, cualesquiera punteros al fragmento de datos a partir de las asignaciones de flujo se pueden modificar en las asignaciones de flujo al valor de desplazamiento de fragmento cambiado, y la entrada 902 se puede eliminar de la tabla de redirección 900. En algunas situaciones, con el tiempo la tabla de redirección 900 se puede vaciar, de esta forma, de las entradas 902.

Las entradas a una tabla de redirección se pueden añadir de varias formas. Por ejemplo, la figura 10 muestra un diagrama de flujo 1000 para almacenar un flujo de datos, de acuerdo con una realización a modo de ejemplo. Realizaciones estructurales y operativas adicionales resultarán evidentes a los expertos en la materia o materias relevantes basándose en el análisis con respecto al diagrama de flujo 1000. El diagrama de flujo 1000 se describe tal como sigue.

El diagrama de flujo 1000 comienza con la etapa 1002. En la etapa 1002, se modifican los contenidos del contenedor de fragmentos. Por ejemplo, en una realización, uno o más fragmentos de datos 322 en el contenedor de fragmentos 304 de la figura 8 pueden ser movidos. Tales fragmentos de datos 322 pueden ser movidos por una tarea de mantenimiento (por ejemplo, el módulo de mantenimiento 106 en la figura 1), tal como un procedimiento de

desfragmentación, un procedimiento de compactación después de una recolección de elementos no usados, u otro procedimiento.

En la etapa 1004, una o más entradas se añaden a la tabla de redirección que indicaba valores de desplazamiento de fragmento cambiados para uno o más fragmentos de datos del contenedor de fragmentos debido a la etapa 1002.

- 5 Por ejemplo, se pueden añadir una o más entradas 902 a la tabla de redirección 900 que se corresponden con los uno o más fragmentos de datos 322 movidos. Por ejemplo, para cada fragmento de datos 322 movido, se puede generar una entrada 902 que indica el valor de identificador local del fragmento de datos 322 movido como el identificador local 904, e indica el nuevo valor de desplazamiento del fragmento de datos 322 movido como el valor de desplazamiento de fragmento cambiado 906.

- 10 En la etapa 1006, la indicación de generación en el encabezado de contenedor de fragmentos se aumenta debido a la etapa 1002. Por ejemplo, en una realización, la indicación de generación de contenedor de fragmentos 804 puede tener un valor inicial de 0, y cada vez que los fragmentos de datos 322 son movidos en el contenedor de fragmentos 304, la indicación de generación de contenedor de fragmentos 804 se puede incrementar para indicar un valor de generación más alto. En otras realizaciones, la indicación de generación de contenedor de fragmentos 804 se puede modificar de otras formas.

- 15 En ese sentido, cuando se busca un fragmento de datos 322 del contenedor de fragmentos 304 de la figura 8 usando el identificador de fragmento de datos que se almacena en la asignación de flujo 310 referente, la indicación de generación de contenedor de fragmentos 804 del contenedor de fragmentos 304 se puede verificar para ver si la generación actual del contenedor de fragmentos 304 es la misma que el valor de generación de contenedor de fragmentos del identificador de fragmento de datos. Si estas son las mismas, el fragmento de datos 322 puede estar ubicado en el desplazamiento que se indica por el valor de desplazamiento de fragmento en el identificador de fragmento de datos. Si no es así, la tabla de redirección 900 se lee para determinar el valor de desplazamiento cambiado del fragmento de datos 322 en el contenedor de fragmentos 304.

- 20 Por ejemplo, la figura 11 muestra un diagrama de bloques de un módulo de rehidratación 1130 que comunica con el contenedor de flujo 302 y el contenedor de fragmentos 304 para rehidratar un flujo de datos de acuerdo con una solicitud de flujo de datos 1110, de acuerdo con una realización a modo de ejemplo. Tal como se muestra en la figura 11, el módulo de rehidratación 1130 incluye un ensamblador de flujos de datos 1102, una unidad de verificación de generación 1106 y una unidad de recuperación de fragmentos de datos 1108. La figura 11 se describe tal como sigue.

- 25 En la figura 11, el ensamblador de flujos de datos 1102 recibe la solicitud de flujo de datos 1110, la cual indica una asignación de flujo, tal como la asignación de flujo 1104 que se almacena en el contenedor de flujo 302, que se corresponde con un flujo de datos que se va a rehidratar. El ensamblador de flujos de datos 1102 procesa la asignación de flujo 1104, generando una solicitud de fragmento de datos 1112 para cada fragmento de datos al que hace referencia la asignación de flujo 1104.

- 30 En una realización, la solicitud de fragmento de datos 1112 generada por el ensamblador de flujos de datos 1102 puede incluir un identificador de fragmento de datos para identificar un fragmento de datos 322 solicitado. Se puede acceder a al contenedor de fragmentos localizado tal como sigue para recuperar fragmentos de datos solicitados.

- 35 Tal como se muestra en la figura 11, la unidad de verificación de generación 1106 recibe la solicitud de fragmento de datos 1112 para el fragmento de datos solicitado. La unidad de verificación de generación 1106 accede al contenedor de fragmentos 304 (que se ha identificado en lo que antecede como que tiene un identificador de contenedor de fragmentos 802 que coincide con el identificador de contenedor de fragmentos del fragmento de datos 322 solicitado). La unidad de verificación de generación 1106 está configurada para comparar la indicación de generación de contenedor de fragmentos 804 para el contenedor de fragmentos 304 con el valor de generación de contenedor de fragmentos para el fragmento de datos 322 solicitado, y para emitir una indicación de coincidencia de generación 1114. Si sus valores no coinciden (por ejemplo, el valor de la indicación de generación de contenedor de fragmentos 804 es más grande que el valor del valor de generación de contenedor de fragmentos para el fragmento de datos 322 solicitado), la indicación de coincidencia de generación 1114 indica que no se halló una coincidencia, y la operación avanza a la etapa 1806. Si sus valores sí coinciden, la indicación de coincidencia de generación 1114 indica que se halló una coincidencia, y se puede seguir una ruta de E/S convencional (u otra ruta) para recuperar el fragmento de datos solicitado.

- 40 Tal como se muestra en la figura 11, la unidad de recuperación de fragmentos de datos 1108 recibe la indicación de coincidencia de generación 1114 y la solicitud de fragmento de datos 1112. Si la indicación de coincidencia de generación 1114 indica que no se halló una coincidencia, la unidad de recuperación de fragmentos de datos 1108 accede a la tabla de redirección 900 para un valor de desplazamiento de fragmento cambiado 906 (la figura 9) en una entrada 902 que tiene un identificador local 904 que coincide con el identificador local del fragmento de datos 322 solicitado. Tal como se muestra en la figura 11, la unidad de recuperación de fragmentos de datos 1108 recibe un segundo valor de desplazamiento de fragmento 1116 que es diferente del primer valor de desplazamiento de fragmento.

Tal como se muestra en la figura 11, la unidad de recuperación de fragmentos de datos 1108 accede al contenedor de fragmentos 304 para un fragmento de datos 322z que está ubicado en el segundo valor de desplazamiento de fragmento 1116. El fragmento de datos 322z es el fragmento de datos 322 solicitado, que ha sido movido en el contenedor de fragmentos 304 al segundo valor de desplazamiento de fragmento 1116.

5 Tal como se muestra en la figura 11, la unidad de recuperación de fragmentos de datos 1108 emite el fragmento de datos 1118, el cual es el fragmento de datos 322z en el ejemplo actual. El fragmento de datos 1118 es recibido por el ensamblador de flujos de datos 1102. De esta forma, el ensamblador de flujos de datos 1102 recibe todos los fragmentos de datos 322 a los que hace referencia la asignación de flujo 1104 a partir de la unidad de recuperación de fragmentos de datos 1108, que se recuperan o bien directamente del contenedor de fragmentos 304 de acuerdo con el valor de desplazamiento de fragmento correspondiente, o bien del contenedor de fragmentos 304 según son redirigidos por la tabla de redirección 900. Tal como se muestra en la figura 11, el ensamblador de flujos de datos 1102 genera un flujo de datos 1120, el cual es la forma rehidratada del flujo de datos solicitado que se indica en la solicitud de flujo de datos 1110. El ensamblador de flujos de datos 1102 ensambla de forma conjunta la totalidad de los fragmentos de datos 322 recibidos tal como se describe en otra parte en el presente documento para formar el flujo de datos 1120.

Se hace notar que el identificador de referencia de asignación de flujo que reside en el punto de repetición de análisis sintáctico de un flujo de datos (por ejemplo, el vínculo de flujo 608a o 608b en la figura 6) puede tener la misma estructura que el identificador de fragmento de datos. Tal como se ha descrito en lo que antecede, una asignación de flujo 310 puede tener forma de un fragmento de datos 322 que contiene metadatos de asignación de flujo en lugar de datos de archivo de usuario final. En ese sentido, el procedimiento para abordar una asignación de flujo 310 puede ser el mismo que abordar un fragmento de datos 322 - ambas técnicas pueden usar la estructura de identificador de fragmento de datos. Un flujo de datos optimizado hace referencia a una asignación de flujo 310 mediante la colocación del identificador de fragmento de datos de la asignación de flujo 310 en el punto de repetición de análisis sintáctico de archivo (unido al flujo de datos/objeto de archivo real). El identificador de asignación de flujo contiene la información de [identificador de contenedor, identificador local, valor de generación, valor de desplazamiento] que se puede usar para localizar (o bien directamente, o bien a través de una tabla de redirección) el fragmento de datos de la asignación de flujo 310 en el interior del contenedor de flujo 302. En ese sentido, en una realización, un formato y distribución de un contenedor de flujo 302 pueden ser esencialmente los mismos que los de un contenedor de fragmentos 304.

30 D. Realizaciones de copias de seguridad de optimización de datos a modo de ejemplo

La realización de una copia de seguridad y la restauración de un sistema de datos que implanta una técnica de optimización de datos son difíciles debido a que se comparten datos entre múltiples flujos de datos en un almacén de fragmentos. En ese sentido, los datos están separados del espacio de nombres de archivo. No obstante, las capacidades de restauración y de copia de seguridad de datos son útiles. Por lo general, las entidades no serían susceptibles de implantar una solución de optimización de datos sin una integración de copias de seguridad de datos eficaz.

En unas realizaciones, se proporcionan varias técnicas de copia de seguridad para entornos de optimización de datos, incluyendo la técnica de copia de seguridad optimizada, de copia de seguridad sin optimizar, de copia de seguridad optimizada a nivel de elemento y de copia de seguridad híbrida. Además, en unas realizaciones, se puede usar una heurística para seleccionar entre diferentes técnicas de copia de seguridad. Por ejemplo, se puede usar una heurística para seleccionar entre una copia de seguridad optimizada y sin optimizar. Unas realizaciones proporcionan sistemas de optimización de datos con unas técnicas de copia de seguridad optimizada, de tal modo que se puede realizar una copia de seguridad de datos en su forma optimizada (por ejemplo, desduplicada). Unas realizaciones pueden posibilitar que la copia de seguridad de datos use menos espacio de medios de copia de seguridad, y se pueden usar para reducir el lapso de tiempo de realización de copia de seguridad, el cual es significativo considerando el crecimiento de datos de año a año. Además, unas realizaciones pueden posibilitar una restauración de datos más rápida a partir de una copia de seguridad (por ejemplo, un RTO (*Recovery Time Objective*, Objetivo de Tiempo de Recuperación) más pequeño).

En unas realizaciones, la copia de seguridad en sistemas de optimización de datos se puede realizar de varias formas. Por ejemplo, la figura 12 muestra un diagrama de flujo 1200 para realizar una copia de seguridad de datos en un sistema de optimización de datos, de acuerdo con una realización a modo de ejemplo. El diagrama de flujo 1200 puede ser realizado por la interfaz de almacén de fragmentos 116 de la figura 1, en una realización. El diagrama de flujo 1200 se describe con respecto a la figura 13 por razones de ilustración. La figura 13 muestra un diagrama de bloques de un sistema de copia de seguridad de datos 1300, de acuerdo con una realización a modo de ejemplo. El diagrama de flujo 1200 puede ser realizado por el sistema de copia de seguridad de datos 1300. Tal como se muestra en la figura 13, el sistema de copia de seguridad de datos 1300 incluye un módulo de copia de seguridad de datos 1302, un almacenamiento de copia de seguridad 1304 y un almacén de fragmentos 1334 que incluye el contenedor de flujo 302 y uno o más contenedores de fragmentos 304 (por ejemplo, los contenedores de fragmentos 304a y 304b). Realizaciones estructurales y operativas adicionales resultarán evidentes a los expertos en la materia o materias relevantes basándose en el análisis con respecto al diagrama de flujo 1200. El diagrama de flujo 1200 se describe tal como sigue.

En la etapa 1202 del diagrama de flujo 1200, una pluralidad de flujos de datos optimizados que están almacenados en un almacén de fragmentos se identifica para copia de seguridad. Por ejemplo, con referencia a la figura 13, el módulo de copia de seguridad de datos 1302 puede recibir una solicitud 1318 para almacenar uno o más flujos de datos optimizados que están almacenados en el almacén de fragmentos 1334 en el almacenamiento de copia de seguridad 1304. La solicitud 1318 puede identificar los uno o más flujos de datos optimizados mediante unos nombres de archivo correspondientes para las estructuras de flujo optimizado (por ejemplo, archivos) que se corresponden con los flujos de datos optimizados. Cada estructura de flujo optimizado hace referencia a un fragmento de asignación de flujo de los fragmentos de asignación de flujo 1316 en el contenedor de flujo 302 que contiene unos metadatos que describen una asignación del flujo de datos optimizado a los fragmentos de datos 322 que se almacenan en uno o más de los contenedores de fragmentos 304, tal como se ha descrito en lo que antecede.

En la etapa 1204, al menos una porción del almacén de fragmentos se almacena en un almacenamiento de copia de seguridad para realizar una copia de seguridad de la pluralidad de flujos de datos optimizados. En respuesta a la solicitud 1318, el módulo de copia de seguridad de datos 1302 puede almacenar al menos una porción del almacén de fragmentos 1334 en el almacenamiento de copia de seguridad 1304 de tal modo que los flujos de datos optimizados que se identifican en la solicitud 1318 se almacenan en el almacenamiento de copia de seguridad 1304. El módulo de copia de seguridad de datos 1302 puede almacenar los flujos de datos optimizados mediante el almacenamiento de una porción del almacén de fragmentos 1334 en el almacenamiento de copia de seguridad 1304 de varias formas. Por ejemplo, en una realización, para cada flujo de datos optimizado, el módulo de copia de seguridad de datos 1302 puede determinar el fragmento de asignación de flujo correspondiente de los fragmentos de asignación de flujo 1316, y los fragmentos de datos 322 a los que hace referencia el fragmento de asignación de flujo, y puede almacenar el fragmento de asignación de flujo y los fragmentos de datos determinados en el almacenamiento de copia de seguridad 1304. En unas realizaciones adicionales, el módulo de copia de seguridad de datos 1302 puede almacenar unas porciones más grandes del almacén de fragmentos 1334 en el almacenamiento de copia de seguridad 1304 que incluyen los fragmentos determinados de los flujos de datos optimizados para realizar una copia de seguridad de los flujos de datos optimizados.

En ese sentido, el módulo de copia de seguridad de datos 1302 se puede configurar para almacenar flujos de datos optimizados de varias formas de acuerdo con la etapa 1204. Por ejemplo, tal como se muestra en la figura 13, el módulo de copia de seguridad de datos 1302 puede incluir un módulo de copia de seguridad de archivos optimizada 1306, un módulo de copia de seguridad de rehidratación 1308, un módulo de copia de seguridad a nivel de elemento 1310 y un módulo de copia de seguridad de identificadores de fragmento de datos 1312. El módulo de copia de seguridad de datos 1302 puede incluir uno o más cualesquiera de los módulos 1306, 1308, 1310 y 1312, en unas realizaciones. Los módulos 1306, 1308, 1310 y 1312 posibilitan que los flujos de datos optimizados se almacenen en el almacenamiento de copia de seguridad 1304 de varias formas. Los módulos 1306, 1308, 1310 y 1312 se describen tal como sigue.

En una realización, una copia de seguridad optimizada (a la que también se hace referencia como “copia de seguridad de flujo de datos optimizado”) se puede realizar para realizar una copia de seguridad de flujos de datos optimizados en una forma optimizada. Por ejemplo, el módulo de copia de seguridad de archivos optimizada 1306 se puede configurar para realizar una copia de seguridad optimizada mediante el almacenamiento de flujos de datos optimizados en el almacenamiento de copia de seguridad 1304 en una forma optimizada. De acuerdo con la copia de seguridad optimizada, uno o más contenedores de fragmentos enteros del almacén de fragmentos 1334 se pueden almacenar para realizar una copia de seguridad de flujos de datos optimizados. Por ejemplo, el módulo de copia de seguridad de archivos optimizada 1306 puede realizar un diagrama de flujo 1400 que se muestra en la figura 14, en una realización. El diagrama de flujo 1400 y el módulo de copia de seguridad de archivos optimizada 1306 se describen tal como sigue.

En la etapa 1402, el almacén de fragmentos se almacena en su totalidad en el almacenamiento de copia de seguridad. En una realización, el módulo de copia de seguridad de archivos optimizada 1306 puede almacenar el almacén de fragmentos 1334 en su totalidad en el almacenamiento de copia de seguridad 1304 de tal modo que se realiza una copia de seguridad de los flujos de datos optimizados que se indican en la solicitud 1318. Como alternativa, el módulo de copia de seguridad de archivos optimizada 1306 puede almacenar uno o más contenedores de fragmentos enteros (por ejemplo, el contenedor de flujo 302 y uno o más de los contenedores de fragmentos 304) del almacén de fragmentos 1334 que incluyen los flujos de datos optimizados solicitados en sus totalidades de tal modo que se realiza una copia de seguridad de los flujos de datos optimizados. Los contenedores de fragmentos a los que se va a realizar una copia de seguridad pueden ser identificados por unos identificadores de contenedor de fragmentos de los identificadores de fragmento para los fragmentos a los que hacen referencia las estructuras de flujo optimizado de los flujos de datos optimizados. Los contenedores de fragmentos/el almacén de fragmentos entero pueden ser almacenados por el módulo de copia de seguridad de archivos optimizada 1306 en el almacenamiento de copia de seguridad 1304 de acuerdo con una operación de almacenamiento 1320.

En la etapa 1404, una pluralidad de códigos auxiliares de metadatos de flujo se almacenan en el almacenamiento de copia de seguridad para la pluralidad de flujos de datos optimizados que establecen un vínculo con unos datos correspondientes en el almacén de fragmentos. Los códigos auxiliares de metadatos de flujo pueden ser recuperados por el módulo de copia de seguridad de archivos optimizada 1306 a partir de otro almacenamiento (por

ejemplo, un almacenamiento primario), tal como un almacenamiento que no sea el almacenamiento de copia de seguridad 1304, y son archivos que establecen un vínculo con unos datos que se almacenan en el almacén de fragmentos 1334 para los flujos de datos optimizados. Por ejemplo, cada código auxiliar de metadatos de flujo puede contener una referencia a su fragmento de asignación de flujo correspondiente de los fragmentos de asignación de flujo 1316 que se pueden usar para reconstruir/volver a ensamblar el flujo de datos correspondiente durante un procedimiento de rehidratación en una forma sin optimizar al combinar cualesquiera fragmentos de datos 322 a los que hace referencia el fragmento de asignación de flujo. En una realización, unos códigos auxiliares de metadatos de flujo que se corresponden con los flujos de datos optimizados que se identifican en la solicitud 1318 son almacenados por el módulo de copia de seguridad de archivos optimizada 1306 en el almacenamiento de copia de seguridad 1304 usando una operación de almacenamiento 1320. Los códigos auxiliares de metadatos de flujo pueden ser almacenados por el módulo de copia de seguridad de archivos optimizada 1306 en el almacenamiento de copia de seguridad 1304 en su formato “en disco”.

En ese sentido, la “imagen de volumen” que proporciona un estado de archivo para los flujos de datos optimizados en el almacenamiento de copia de seguridad 1304 es un reflejo del almacén de fragmentos 1334. La combinación de contenedores de fragmentos almacenados y códigos auxiliares de metadatos de flujo almacenados en el almacenamiento de copia de seguridad 1304 proporciona un almacenamiento completo de los flujos de datos optimizados, en forma optimizada (no vuelta a ensamblar). No es necesario que se almacenen metadatos adicionales.

Una copia de seguridad optimizada (por ejemplo, de acuerdo con el diagrama de flujo 1400 y el módulo de copia de seguridad de archivos optimizada 1306) se puede realizar en cualquier situación para realizar una copia de seguridad de flujos de datos optimizados. Se puede desear la realización de una copia de seguridad optimizada si se está considerando la mayor parte del volumen al que se le ha realizado una copia de seguridad (un área de almacenamiento accesible en un almacenamiento de copia de seguridad). Algo de “desperdicio” es posible debido a que cuando se almacenan unos contenedores de fragmentos completos debido a que los fragmentos de datos a los que no hacen referencia los flujos de datos optimizados que se identifican en la solicitud 1318, pero a los que hacen referencia otros flujos de datos optimizados, se pueden incluir en los contenedores de fragmentos. En ese sentido, se puede realizar una copia de seguridad de más datos en conjunto de lo que es necesario. Si se está considerando relativamente poco del volumen al que se le ha realizado una copia de seguridad (por ejemplo, un pequeño número de los fragmentos de datos que se incluyen en los contenedores de fragmentos con copia de seguridad están asociados con los flujos de datos optimizados que se identifican en la solicitud 1318), se puede desear otra técnica de copia de seguridad (por ejemplo, una técnica de copia de seguridad no optimizada que se describe en lo sucesivo) que no almacene contenedores de fragmentos enteros.

La restauración de flujos de datos optimizados a los cuales se les ha realizado una copia de seguridad de acuerdo con la copia de seguridad optimizada comporta restaurar los archivos de contenedor de almacén de fragmentos y los códigos auxiliares de metadatos de flujo, de tal modo que la restauración es una restauración optimizada. Un almacén optimizado de este tipo tiene un tamaño relativamente más pequeño y, por lo tanto, es relativamente más rápido, y se consumen menos recursos de E/S. Unas técnicas para la restauración selectiva de flujos de datos optimizados a partir de una copia de seguridad optimizada se describen adicionalmente en lo sucesivo en la siguiente subsección.

En ese sentido, unas realizaciones para una copia de seguridad optimizada pueden proporcionar varios beneficios. Por ejemplo, una copia de seguridad optimizada puede dar como resultado unas copias de seguridad de tamaño de almacenamiento más pequeño, las cuales ahorran espacio de medios de copia de seguridad (por ejemplo, en el almacenamiento de copia de seguridad 1304). Esto puede ser útil para las soluciones de copia de seguridad basadas en discos. En algunos casos, los ahorros de espacio de almacenamiento en el almacenamiento de copia de seguridad pueden ser más significativos y rentables que los ahorros en el almacenamiento primario. Una copia de seguridad optimizada puede dar como resultado un tiempo más rápido usado para realizar la copia de seguridad. El tiempo de ejecución de copia de seguridad se puede reducir, lo cual es significativo considerando el crecimiento en la cantidad de datos que se están almacenando de año a año. Debido al crecimiento en la cantidad de almacenamiento de datos, los usuarios pueden tener dificultades con la realización de copias de seguridad frecuentes (por ejemplo, a diario), debido a que el tiempo de ejecución de copia de seguridad crece junto con la cantidad de datos. Las técnicas de copia de seguridad optimizada pueden ayudar con la reducción del tiempo de ejecución de copia de seguridad. Además, una copia de seguridad optimizada puede acortar el RTO, conduciendo a unas restauraciones más rápidas. El RTO es una importante medida de una solución de copia de seguridad/restauración.

En otra realización, una copia de seguridad sin optimizar (a la que también se hace referencia como “copia de seguridad de flujo de datos no optimizado”) se puede realizar para almacenar flujos de datos optimizados en una forma sin optimizar o no optimizada. De acuerdo con la copia de seguridad sin optimizar, los flujos de datos optimizados que se designan para un almacenamiento de copia de seguridad se pueden rehidratar antes de almacenarse. Por ejemplo, en una realización, el módulo de copia de seguridad de rehidratación 1308 se puede configurar para realizar una copia de seguridad sin optimizar mediante la rehidratación de flujos de datos optimizados y el almacenamiento de los flujos de datos rehidratados en el almacenamiento de copia de seguridad 1304. El módulo de copia de seguridad de rehidratación 1308 puede realizar un diagrama de flujo 1500 que se

muestra en la figura 15, en una realización. El diagrama de flujo 1500 y el módulo de copia de seguridad de rehidratación 1308 se describen tal como sigue.

En la etapa 1502 del diagrama de flujo 1500, cada flujo de datos optimizado se rehidrata en un flujo de datos no optimizado correspondiente que incluye cualesquiera fragmentos de datos a los que hacen referencia los metadatos de flujo optimizado correspondientes. El módulo de copia de seguridad de rehidratación 1308 puede rehidratar flujos de datos optimizados de cualquier forma, incluyendo tal como se ha descrito en lo que antecede para el módulo de rehidratación 702 (la figura 7). Por ejemplo, haciendo referencia a la figura 13, el módulo de copia de seguridad de rehidratación 1308 puede acceder al contenedor de flujo 302 para metadatos de flujo optimizado en forma de fragmentos de asignación de flujo que se corresponden con los flujos de datos optimizados que se identifican en la solicitud 1318, o en otra forma (por ejemplo, una tabla global, una base de datos, etc.). Tal como se muestra en la figura 13, el módulo de copia de seguridad de rehidratación 1308 puede generar uno o más accesos de contenedor de flujo 1322 para los fragmentos de asignación de flujo deseados. Los accesos de contenedor de flujo 1322 pueden identificar los fragmentos de asignación de flujo deseados mediante los identificadores de asignación de flujo correspondientes. En respuesta a los accesos de contenedor de flujo 1322, el módulo de copia de seguridad de rehidratación 1308 recibe los fragmentos de asignación de flujo 1324 a partir del contenedor de flujo 302 que se corresponden con los flujos de datos optimizados que se identifican en la solicitud 1318. El módulo de copia de seguridad de rehidratación 1308 puede regenerar o "rehidratar" cada flujo de datos optimizado de acuerdo con la asignación de flujo del fragmento de asignación de flujo recuperado correspondiente. En el ejemplo de la figura 13, cada asignación de flujo incluye punteros (por ejemplo, el identificador de fragmento de datos 404 de la figura 4) a cada uno de los fragmentos de datos que se incluyen en el flujo de datos optimizado correspondiente, el cual se puede incluir en los contenedores de fragmentos 304a y/o 304b. El módulo de copia de seguridad de rehidratación 1308 usa los punteros para recuperar cada uno de los fragmentos de datos 322 a los que se hace referencia. Por ejemplo, el módulo de copia de seguridad de rehidratación 1308 puede generar uno o más de los accesos de contenedor de fragmentos 1326 y/o 1330 para obtener los fragmentos de datos 322 a partir del primer y/o el segundo contenedores de fragmentos 304a y 304b, de forma respectiva. En respuesta a los accesos de contenedor de fragmentos 1326 y/o 1330, el módulo de copia de seguridad de rehidratación 1308 recibe los fragmentos de datos 1328 y/o 1332 a partir de los contenedores de fragmentos 302a y 302b que se corresponden con los flujos de datos optimizados que se identifican en la solicitud 1318.

El módulo de copia de seguridad de rehidratación 1308 puede usar los desplazamientos de flujo de datos 402 que se incluyen en las asignaciones de flujo recuperadas (por ejemplo, la asignación de flujo 310 de la figura 3) para disponer los fragmentos de datos recuperados en el orden apropiado para regenerar cada flujo de datos en una forma no optimizada (por ejemplo, como flujos de datos no optimizados).

En la etapa 1504, cada flujo de datos no optimizado se almacena en el almacenamiento de copia de seguridad. Por ejemplo, tal como se muestra en la figura 13, los flujos de datos no optimizados pueden ser almacenados por el módulo de copia de seguridad de rehidratación 1308 en el almacenamiento de copia de seguridad 1304 de acuerdo con una operación de almacenamiento 1320. En ese sentido, cuando se desea restaurar un flujo de datos a partir del almacenamiento de copia de seguridad 1304, el flujo de datos no optimizado que está almacenado en el almacenamiento de copia de seguridad 1304 se puede recuperar.

Por lo tanto, una copia de seguridad sin optimizar (por ejemplo, de acuerdo con el diagrama de flujo 1500 y el módulo de copia de seguridad de rehidratación 1308) se puede realizar en cualquier situación para realizar una copia de seguridad de flujos de datos optimizados. Se puede desear que una copia de seguridad sin optimizar se realice si fragmentos de datos de los flujos de datos optimizados rellenan una porción relativamente menor del espacio de almacenamiento del almacén de fragmentos 1334. Por lo tanto, en lugar de almacenar unos contenedores de fragmentos enteros en el almacenamiento de copia de seguridad 1304, lo cual puede incluir almacenar un gran número de fragmentos de datos no relacionados con los flujos de datos optimizados que se identifican en la solicitud 1318, los flujos de datos optimizados específicos se pueden rehidratar y almacenar en el almacenamiento de copia de seguridad 1304. En ese sentido, los medios de copia de seguridad almacenan los flujos de datos en su forma original no optimizada, y se puede ahorrar espacio de almacenamiento de copia de seguridad al evitar almacenar fragmentos de datos no relacionados.

Unas realizaciones para una copia de seguridad sin optimizar pueden proporcionar varios beneficios. Por ejemplo, una copia de seguridad sin optimizar, la cual usa una copia de seguridad más selectiva (y posibilita una restauración selectiva), es relativamente fácil de implementar. La restauración de los flujos de datos con copia de seguridad no depende de la técnica de optimización de datos que es usada por un sistema de almacenamiento, debido a que se realiza una copia de seguridad de los flujos de datos en una forma no optimizada. En ese sentido, los flujos de datos se pueden restaurar en cualquier parte y se puede acceder a los mismos sin depender de una solución de optimización de datos instalada y funcional.

Debido al procedimiento de rehidratación, una copia de seguridad sin optimizar puede ser relativamente lenta, teniendo un impacto de rendimiento sobre el módulo de copia de seguridad de datos 1302. La rehidratación de datos optimizados es más lenta que una lectura de datos ordinaria debido a la descompresión de los datos optimizados que se realiza, y también potencialmente debido a la fragmentación de datos. Además, debido a que se realiza una copia de seguridad de los flujos de datos en una forma no optimizada, la cantidad total de datos de los cuales se

realiza una copia de seguridad puede ser grande debido a que las ventajas de la deduplicación no se encuentran presentes. La cantidad total de datos es potencialmente una cantidad mayor que el volumen al cual se le está realizando una copia de seguridad (debido a que el volumen se optimiza mientras que los datos de copia de seguridad no lo están). En algunos casos, puede que no sea posible realizar una copia de seguridad de la totalidad de los datos debido a las limitaciones de tamaño de almacenamiento de copia de seguridad. Por lo tanto, una copia de seguridad sin optimizar se puede seleccionar para su uso en situaciones de copia de seguridad particulares. Se describen adicionalmente los sucesivos ejemplos de selección de una copia de seguridad sin optimizar frente a otras técnicas de copia de seguridad.

En otra realización, una copia de seguridad a nivel de elemento se puede realizar para almacenar flujos de datos optimizados en una forma optimizada a nivel de elemento. De acuerdo con la copia de seguridad a nivel de elemento, los flujos de datos optimizados que se designan para un almacenamiento de copia de seguridad se preparan para su almacenamiento en una forma optimizada. Por ejemplo, para un flujo de datos optimizado particular, se determinan los fragmentos de datos a los que hace referencia el fragmento de asignación de flujo de los flujos de datos optimizados. Ninguno de los fragmentos de datos a los que se hace referencia que ya se han almacenado en un almacenamiento de copia de seguridad se recupera del almacén de fragmentos. Cualquiera de los fragmentos de datos a los que se hace referencia que no están almacenados aún en un almacenamiento de copia de seguridad se recupera del almacén de fragmentos, y se almacena en un almacenamiento de copia de seguridad. Por ejemplo, en una realización, el módulo de copia de seguridad a nivel de elemento 1310 se puede configurar para realizar una copia de seguridad a nivel de elemento al almacenar en el almacenamiento de copia de seguridad 1304 los fragmentos de asignación de flujo y los fragmentos de datos a los que se hace referencia que no están almacenados aún en el almacenamiento de copia de seguridad 1304. Por ejemplo, el módulo de copia de seguridad a nivel de elemento 1310 puede realizar un diagrama de flujo 1600 que se muestra en la figura 16, en una realización. El diagrama de flujo 1600 se puede realizar para cada flujo de datos optimizado que se identifica en la solicitud 1318. El diagrama de flujo 1600 y el módulo de copia de seguridad a nivel de elemento 1310 se describen tal como sigue.

En la etapa 1602 del diagrama de flujo 1600, se recibe un primer flujo de datos optimizado identificado para copia de seguridad. Por ejemplo, la solicitud 1318 puede identificar un flujo de datos optimizado. Para el flujo de datos optimizado, el módulo de copia de seguridad a nivel de elemento 1310 puede recuperar los metadatos de flujo optimizado (por ejemplo, el fragmento de asignación de flujo 1324 a partir del contenedor de flujo 302) y puede recuperar cualesquiera fragmentos de datos 322 a los que hacen referencia los metadatos de flujo optimizado a partir de los contenedores de fragmentos 304. Por ejemplo, el módulo de copia de seguridad a nivel de elemento 1310 puede generar un acceso de contenedor de flujo 1322 al contenedor de flujo 302 para recuperar el fragmento de asignación de flujo deseado como el fragmento de asignación de flujo 1324, y puede generar uno o más de los accesos de contenedor de fragmentos 1326 y/o 1330 para obtener los fragmentos de datos 322 a los que se hace referencia a partir del primer y/o el segundo contenedores de fragmentos 304a y 304b como los fragmentos de datos 1328 y/o 1332.

En la etapa 1604, se determina o determinan uno o más fragmentos de datos a los que hacen referencia los metadatos de flujo optimizado del primer flujo de datos optimizado y no almacenados aún en el almacenamiento de copia de seguridad. En una realización, el módulo de copia de seguridad a nivel de elemento 1310 puede comparar los fragmentos de datos a los que hacen referencia los metadatos del fragmento de asignación de flujo con cualesquiera fragmentos de datos ya almacenados en el almacenamiento de copia de seguridad 1304 para determinar si cualquiera de los fragmentos de datos a los que se hace referencia ya están almacenados en el almacenamiento de copia de seguridad 1304. El módulo de copia de seguridad a nivel de elemento 1310 puede realizar la comparación de cualquier forma, incluyendo mediante la comparación de valores de troceo de los fragmentos de datos a los que se hace referencia con valores de troceo de los fragmentos de datos que se almacenan en el almacenamiento de copia de seguridad 1304 (por ejemplo, usando un índice de troceo, etc.), puede comparar identificadores de fragmento de datos para los fragmentos de datos a los que se hace referencia con identificadores de fragmento de datos de los fragmentos de datos que se almacenan en el almacenamiento de copia de seguridad 1304, o puede hacer la comparación de acuerdo con otra técnica. El módulo de copia de seguridad a nivel de elemento 1310 puede mantener un seguimiento de los fragmentos de datos a los que se hace referencia que se determina que no están almacenados aún en el almacenamiento de copia de seguridad 1304, tal como al mantener una lista u otra estructura de datos (por ejemplo, en memoria).

En la etapa 1606, los metadatos de flujo optimizado del primer flujo de datos optimizado se almacenan en el almacenamiento de copia de seguridad. Por ejemplo, el módulo de copia de seguridad a nivel de elemento 1310 puede almacenar el fragmento de asignación de flujo que se recupera del contenedor de flujo 302 para el flujo de datos optimizado en el almacenamiento de copia de seguridad 1304 (por ejemplo, usando una operación de almacenamiento 1320).

En la etapa 1608, los uno o más fragmentos de datos determinados se almacena o almacenan en el almacenamiento de copia de seguridad. El módulo de copia de seguridad a nivel de elemento 1310 puede almacenar los fragmentos de datos que se recuperan de los contenedores de fragmentos 302a y/o 302b para el flujo de datos optimizado, y que se determina en la etapa 1604 que no están almacenados aún en el almacenamiento de copia de seguridad 1304, en el almacenamiento de copia de seguridad 1304 (por ejemplo, usando una operación de

almacenamiento 1320). En ese sentido, cuando se desea restaurar el flujo de datos a partir del almacenamiento de copia de seguridad 1304, el fragmento de asignación de flujo y los fragmentos de datos que se almacenan en el almacenamiento de copia de seguridad 1304 para el flujo de datos optimizado se pueden recuperar del almacenamiento de copia de seguridad 1304, y rehidratarse para formar el flujo de datos. En una realización, se puede realizar una copia de seguridad de la estructura de flujo optimizado (por ejemplo, el código auxiliar de metadatos de flujo que se ha descrito en lo que antecede con respecto a la etapa 1404 del diagrama de flujo 1400 (la figura 14) en el almacenamiento de copia de seguridad 1304, y se puede recuperar del almacenamiento de copia de seguridad 1304 como un punto de repetición de análisis sintáctico para rehidratar el flujo de datos.

Por lo tanto, una copia de seguridad a nivel de elemento (por ejemplo, de acuerdo con el diagrama de flujo 1600 y el módulo de copia de seguridad a nivel de elemento 1310) se puede realizar en cualquier situación para realizar una copia de seguridad de flujos de datos optimizados. De acuerdo con la copia de seguridad a nivel de elemento, los fragmentos de datos desduplicados están asociados con cada flujo de datos del cual se realiza una copia de seguridad al tiempo que se mantiene la copia de seguridad en una forma optimizada (por ejemplo, sin rehidratación, se realiza una copia de seguridad de cada fragmento de datos una vez). No se realiza una copia de seguridad de los contenedores de almacén de fragmentos completos.

En una realización, una API (*application programming interface*, interfaz de programación de aplicaciones) de copia de seguridad (y, opcionalmente, de restauración) se puede implementar mediante el módulo de copia de seguridad a nivel de elemento 1310. Por ejemplo, una sesión de copia de seguridad se puede definir para realizar una copia de seguridad de un primer archivo y un segundo archivo que se almacenan en el almacén de fragmentos 1334 en una forma optimizada. En el presente ejemplo, se supone que el primer archivo incluye los fragmentos de datos 322a y 322b del contenedor de fragmentos 302a y el fragmento de datos 322d del contenedor de fragmentos 304b, y el segundo archivo incluye los fragmentos de datos 322a - 322c del contenedor de fragmentos 302a y los fragmentos de datos 322d y 322e del contenedor de fragmentos 304b. Se puede llamar a la API de copia de seguridad para realizar una copia de seguridad del primer archivo. En respuesta, la API devolverá el fragmento de asignación de flujo para el primer archivo y los fragmentos de datos 322a, 322b y 322d. El fragmento de asignación de flujo y los fragmentos de datos 322a, 322b y 322d devueltos se almacenan en el almacenamiento de copia de seguridad 1304 para el primer archivo. Se puede llamar a la API de copia de seguridad posteriormente para realizar una copia de seguridad del segundo archivo. En respuesta, la API devolverá el fragmento de asignación de flujo para el segundo archivo y los fragmentos de datos 322c y 322e. Esto es debido a que la API determinará que los fragmentos de datos 322a, 322b y 322d del segundo archivo ya están almacenados en el almacenamiento de copia de seguridad 1304 (debido al primer archivo al cual se le está realizando una copia de seguridad en el almacenamiento de copia de seguridad 1304). En ese sentido, el fragmento de asignación de flujo y los fragmentos de datos 322c y 322e devueltos se almacenan en el almacenamiento de copia de seguridad 1304 para el segundo archivo.

Un módulo de restauración para restaurar flujos de datos a los que se les ha realizado una copia de seguridad de acuerdo con la copia de seguridad a nivel de elemento puede usar una API similar, en el que los metadatos de flujo optimizado ayudan a la API de módulo de restauración a apuntar a los fragmentos de datos a los que se hace referencia. Una copia de seguridad a nivel de elemento posibilita una copia de seguridad optimizada (en cuanto al tamaño) para una copia de seguridad tanto completa como selectiva (debido a que la granularidad de la copia de seguridad es una granularidad de flujo de datos/archivos). No obstante, una copia de seguridad a nivel de elemento puede ser relativamente lenta y compleja, y puede que no funcione bien con las técnicas de copia de seguridad a nivel de bloque.

En aún otra realización, la copia de seguridad de identificador de fragmento de datos se puede realizar para almacenar flujos de datos optimizados en otro tipo de forma optimizada. De acuerdo con la copia de seguridad de identificador de fragmento de datos, los identificadores de fragmento de datos se determinan para flujos de datos optimizados a los que se va a realizar una copia de seguridad. Los identificadores de fragmento de datos se almacenan en un almacenamiento de copia de seguridad, y los contenedores de fragmentos que almacenan los fragmentos de datos a los que se hace referencia se almacenan en un almacenamiento de copia de seguridad. Por ejemplo, en una realización, el módulo de copia de seguridad de identificadores de fragmento de datos 1312 se puede configurar para realizar la copia de seguridad de identificador de fragmento de datos al almacenar en el almacenamiento de copia de seguridad 1304 los identificadores de fragmento de datos a los que hacen referencia los metadatos de flujo optimizado de los flujos de datos, y los contenedores de fragmentos que almacenan los fragmentos que son identificados por los identificadores de fragmento de datos asociados. Por ejemplo, el módulo de copia de seguridad de identificadores de fragmento de datos 1312 puede realizar un diagrama de flujo 1700 que se muestra en la figura 17, en una realización. El diagrama de flujo 1700 y el módulo de copia de seguridad de identificadores de fragmento de datos 1312 se describen tal como sigue.

En la etapa 1702 del diagrama de flujo 1700, los metadatos de flujo optimizado de cada flujo de datos optimizado se analizan para determinar un al menos un identificador de fragmento de datos correspondiente para el al menos un fragmento de datos al que hacen referencia los metadatos de flujo optimizado. Por ejemplo, en una realización, para cada flujo de datos optimizado que se identifica en la solicitud 1318, el módulo de copia de seguridad de identificadores de fragmento de datos 1312 puede recuperar los metadatos de flujo optimizado correspondientes en forma de un fragmento de asignación de flujo a partir del contenedor de flujo 302. Además, el módulo de copia de seguridad de identificadores de fragmento de datos 1312 puede analizar los metadatos del fragmento de asignación

de flujo recuperado para determinar los identificadores de fragmento de datos para cualesquiera fragmentos de datos 322 a los que hacen referencia los metadatos del fragmento de asignación de flujo a partir de los contenedores de fragmentos 304. Los identificadores de fragmento de datos se incluyen en los metadatos (por ejemplo, los identificadores de fragmento de datos 404 que se incluyen en los metadatos 400 que se muestran en la figura 4). El módulo de copia de seguridad de identificadores de fragmento de datos 1312 puede hacer referencia a una tabla de redirección (por ejemplo, la tabla de redirección 900 de la figura 9) para asignar valores de desplazamiento de fragmento para uno o más de los identificadores de fragmento de datos para actualizar valores de desplazamiento de fragmento si uno o más fragmentos de datos a los que se hace referencia fueron movidos en los contenedores de fragmentos 304.

En la etapa 1704, una estructura de flujo optimizado para cada flujo de datos optimizado se almacena en el almacenamiento de copia de seguridad con el al menos un identificador de fragmento de datos correspondiente. Por ejemplo, el módulo de copia de seguridad de identificadores de fragmento de datos 1312 puede almacenar las estructuras de flujo optimizado de cada uno de los flujos de datos optimizados en el almacenamiento de copia de seguridad 1304 con los identificadores de fragmento de datos correspondientes que se determinan en la etapa 1702 (por ejemplo, usando una operación de almacenamiento 1320). Los identificadores de fragmento se pueden almacenar en asociación con las estructuras de flujo optimizado de cualquier forma, incluyendo almacenarse externos a, o dentro de, las estructuras de flujo optimizado correspondientes en el almacenamiento de copia de seguridad 1304.

En la etapa 1706, uno o más contenedores de fragmentos del almacén de fragmentos que contienen los flujos de datos optimizados se almacena o almacenan en el almacenamiento de copia de seguridad. En una realización, el módulo de copia de seguridad de identificadores de fragmento de datos 1312 puede almacenar el contenedor de flujo 302 en el almacenamiento de copia de seguridad 1304 de tal modo que se realiza una copia de seguridad de los fragmentos de asignación de flujo de los flujos de datos optimizados (por ejemplo, usando una operación de almacenamiento 1320). Además, el módulo de copia de seguridad de identificadores de fragmento de datos 1312 puede almacenar los uno o más de los contenedores de fragmentos 304 que almacenan cualesquiera fragmentos de datos 322 a los que hacen referencia los fragmentos de asignación de flujo de los flujos de datos optimizados de tal modo que se realiza una copia de seguridad de los fragmentos de datos a los que se hace referencia (por ejemplo, usando una operación de almacenamiento 1320). En una realización, el módulo de copia de seguridad de identificadores de fragmento de datos 1312 puede almacenar el almacén de fragmentos 1334 en su totalidad en el almacenamiento de copia de seguridad 1304 de tal modo que se realiza una copia de seguridad de todos los fragmentos de los flujos de datos optimizados.

En ese sentido, la copia de seguridad de identificador de fragmento de datos es similar a la copia de seguridad optimizada que se ha descrito en lo que antecede, pero las ubicaciones de todos los fragmentos de datos (fragmentos de datos) en sus archivos de contenedor respectivos también se almacenan (mediante identificadores de fragmento de datos) para proporcionar un “plan de restauración” en el instante de realización de copia de seguridad. Un beneficio de la copia de seguridad de identificador de fragmento de datos es que una aplicación de copia de seguridad (por ejemplo, el módulo de copia de seguridad de datos 1302) puede almacenar en su catálogo el “plan de restauración” y conocer por adelantado qué otros archivos (por lo tanto, qué medios de copia de seguridad) son necesarios para cada escenario de restauración, incluyendo una restauración selectiva. De forma similar a la copia de seguridad sin optimizar que se ha descrito en lo que antecede, la copia de seguridad de identificador de fragmento de datos puede ser relativamente lenta y compleja, y puede usar una API de copia de seguridad que se implementa mediante el módulo de copia de seguridad de identificadores de fragmento de datos 1312. Además, la copia de seguridad de identificador de fragmento de datos rompe la modularidad del almacén de fragmentos y expone la implementación interna de almacén de fragmentos.

En ese sentido, la copia de seguridad optimizada, la copia de seguridad sin optimizar, la copia de seguridad a nivel de elemento y la copia de seguridad de identificador de fragmento de datos son, cada una, unas realizaciones de copia de seguridad que se pueden implementar mediante el módulo de copia de seguridad de datos 1302 de la figura 13. Además, el módulo de copia de seguridad de datos 1302 puede implementar unas realizaciones de copia de seguridad que son combinaciones de una o más de la copia de seguridad optimizada, la copia de seguridad sin optimizar, la copia de seguridad a nivel de elemento y la copia de seguridad de identificador de fragmento de datos.

Por ejemplo, en una realización, el módulo de copia de seguridad de datos 1302 puede implementar una combinación de una copia de seguridad optimizada y una copia de seguridad sin optimizar. En una realización, se puede seleccionar una copia de seguridad optimizada para realizarse cuando se realiza una copia de seguridad de volumen completo. Si se va a realizar una copia de seguridad de un único flujo de datos optimizado, se puede realizar una copia de seguridad sin optimizar. Para números de flujos de datos optimizados entre uno y todos los flujos de datos optimizados, se puede seleccionar una copia de seguridad optimizada o una copia de seguridad sin optimizar. Por ejemplo, se puede alternar entre una copia de seguridad optimizada o una copia de seguridad sin optimizar basándose en heurística.

Por ejemplo, en una realización, el módulo de copia de seguridad de datos 1302 se puede configurar para realizar una copia de seguridad de flujos de datos optimizados de acuerdo con un procedimiento que se muestra en la figura 18. La figura 18 muestra un diagrama de flujo 1800 que proporciona un procedimiento para seleccionar y ejecutar

una técnica de copia de seguridad basándose en heurística, de acuerdo con una realización a modo de ejemplo. El diagrama de flujo 1800 se describe tal como sigue.

En la etapa 1802 del diagrama de flujo 1800, una técnica de copia de seguridad se selecciona basándose en heurística. Por ejemplo, tal como se muestra en la figura 13, el módulo de copia de seguridad de datos 1302 puede incluir opcionalmente un módulo de heurística 1314. El módulo de heurística 1314 está configurado para determinar una heurística que se usa para seleccionar una técnica de copia de seguridad que se va a usar para realizar una copia de seguridad de flujos de datos optimizados.

En la etapa 1804, se realiza la técnica de copia de seguridad para realizar una copia de seguridad de la pluralidad de flujos de datos optimizados en el almacenamiento de copia de seguridad. Basándose en la heurística que es determinada por el módulo de heurística 1314, uno del módulo de copia de seguridad de archivos optimizada 1306 o el módulo de copia de seguridad de rehidratación 1308 se puede seleccionar para realizar una copia de seguridad de los flujos de datos optimizados (por ejemplo, tal como se ha descrito en lo que antecede con respecto a las figuras 14 y 15, de forma respectiva). Por ejemplo, en una realización, el módulo de heurística 1314 puede proporcionar una señal de habilitación al uno del módulo de copia de seguridad de archivos optimizada 1306 o el módulo de copia de seguridad de rehidratación 1308 que se selecciona para realizar una copia de seguridad de los flujos de datos optimizados. La señal de habilitación habilita el uno del módulo de copia de seguridad de archivos optimizada 1306 o el módulo de copia de seguridad de rehidratación 1308 para realizar una copia de seguridad de los flujos de datos optimizados de acuerdo con su técnica de copia de seguridad correspondiente.

Por ejemplo, en una realización, el módulo de heurística 1314 puede implementar una heurística relativamente simple basándose en un “modo de exclusión” y un “modo de inclusión”. De acuerdo con el “modo de exclusión”, un usuario puede seleccionar un volumen de almacén de fragmentos y puede excluir de la copia de seguridad algunas carpetas/archivos. En ese sentido, el módulo de heurística 1314 puede determinar que los flujos de datos optimizados se seleccionaron para copia de seguridad de acuerdo con un modo de exclusión. En tal caso, la técnica de copia de seguridad de flujo de datos optimizado es seleccionada por el módulo de heurística 1314 debido al modo de exclusión determinado. Esto es debido a que el “modo de exclusión” por lo general tiene en consideración la mayor parte del volumen de almacén de fragmentos. En ese sentido, es una mejor compensación recíproca realizar una copia de seguridad de los flujos de datos seleccionados (en su forma optimizada) más todos los archivos de contenedor de almacén de fragmentos, incluso a pesar de que los archivos de contenedor pueden incluir algunos fragmentos a los que hacen referencia unos flujos de datos optimizados que no sean los que fueron seleccionados por el usuario.

De acuerdo con el “modo de inclusión”, un usuario puede seleccionar relativamente pocas carpetas/archivos para copia de seguridad. En ese sentido, el módulo de heurística 1314 puede determinar que los flujos de datos optimizados se seleccionaron para copia de seguridad de acuerdo con un modo de inclusión. En tal caso, una técnica de copia de seguridad de flujo de datos no optimizado es seleccionada por el módulo de heurística 1314 debido al modo de inclusión determinado. Esto es debido a que el “modo de inclusión” por lo general tiene en consideración una porción relativamente pequeña del volumen de almacén de fragmentos. En ese sentido, es una mejor compensación recíproca realizar una copia de seguridad solo de los archivos seleccionados en su forma no optimizada. En tal caso, no hay necesidad alguna de realizar una copia de seguridad de los archivos de contenedor de almacén de fragmentos.

En otra realización, el módulo de heurística 1314 puede implementar una heurística relativamente avanzada. Una compensación recíproca en la técnica de copia de seguridad se puede basar en un diferencial entre el tamaño optimizado y el tamaño lógico de archivos dentro de un ámbito de copia de seguridad. Un desperdicio de espacio de almacenamiento se puede determinar como el espacio de almacenamiento en los archivos de contenedor de almacén de fragmentos que es consumido por fragmentos a los que hacen referencia solo unos flujos de datos optimizados que no se incluyen en el ámbito de copia de seguridad. En ese sentido, el módulo de heurística 1314 puede determinar (por ejemplo, calcular) una cantidad de espacio de almacenamiento desperdiciado basándose en los tamaños de archivo en el volumen de almacén de fragmentos y basándose en estadísticas de almacén de fragmentos. En una realización, se pueden realizar algunas suposiciones a menos que el almacén de fragmentos esté configurado para notificar con precisión a cuántos fragmentos hace referencia un grupo dado de flujos de datos. Si el almacén de fragmentos es capaz de notificar un número de fragmentos a los que hace referencia un grupo designado de flujos de datos optimizados, el módulo de heurística 1314 puede determinar la cantidad de espacio de almacenamiento desperdiciado como el espacio relleno por todos los fragmentos menos el espacio relleno por fragmentos a los que hacen referencia unos flujos de datos optimizados que se designan para copia de seguridad. Basándose en la cantidad de espacio de almacenamiento desperdiciado (por ejemplo, en relación con el espacio de almacenamiento total), el módulo de heurística 1314 puede seleccionar una copia de seguridad optimizada o una copia de seguridad sin optimizar.

Por ejemplo, haciendo referencia a la figura 13, en una realización, el módulo de heurística 1314 puede determinar una cantidad de espacio en el almacén de fragmentos 1334 que es consumida por los fragmentos de datos 322 a los que no hacen referencia los metadatos de fragmento de asignación de flujo de cualquiera de la pluralidad de flujos de datos optimizados que se identifican en la solicitud 1318. El módulo de heurística 1314 puede seleccionar la técnica de copia de seguridad para que sea la técnica de copia de seguridad de flujo de datos optimizado si la

cantidad determinada de espacio en el almacén de fragmentos 1334 que es consumida por los fragmentos de datos a los que no se hace referencia es menor que un umbral previamente determinado (por ejemplo, un 50 % u otro porcentaje de espacio de almacenamiento total del almacén de fragmentos 1334). El módulo de heurística 1314 puede seleccionar la técnica de copia de seguridad para que sea una técnica de copia de seguridad de flujo de datos no optimizado si la cantidad determinada de espacio en el almacén de fragmentos 1334 que es consumida por los fragmentos de datos a los que no se hace referencia es más grande que el umbral previamente determinado.

En un ejemplo, el módulo de heurística 1314 puede realizar el siguiente análisis para establecer una técnica de copia de seguridad. Dado un ámbito de copia de seguridad el cual incluye una o más raíces de espacio de nombres, en el que una raíz de espacio de nombres es una carpeta que incluye subcarpetas de forma recursiva, se pueden definir los siguientes parámetros de heurística:

Ámbito = todos los archivos en un ámbito de copia de seguridad, lo que quiere decir todos los archivos bajo las carpetas seleccionadas para copia de seguridad,

A = tamaño lógico de todos los archivos en el ámbito, en el que un tamaño lógico del archivo es el tamaño del archivo antes de que se deduplicara el archivo (A es la suma de todos los tamaños lógicos del archivo en el ámbito),

B = tamaño en disco de todos los archivos en el ámbito, en el que el tamaño en disco de un archivo es el tamaño real que llena el archivo optimizado en el disco después de la deduplicación del archivo, y

C = tamaño del almacén de fragmentos, el cual es la suma de los tamaños de todos los archivos de contenedor de almacén de fragmentos.

Considerando estas definiciones, si la aplicación de copia de seguridad usa una copia de seguridad optimizada, entonces el tamaño de copia de seguridad total en los medios de copia de seguridad es $B + C$. Si la aplicación de copia de seguridad usa una copia de seguridad sin optimizar, entonces el tamaño de copia de seguridad total en los medios de copia de seguridad es A.

Por lo tanto, el módulo de heurística 1314 puede calcular unos valores para A y $B + C$ para ayudar a decidir en la selección entre una copia de seguridad optimizada o sin optimizar después de que se haya determinado el ámbito de archivos de copia de seguridad. Si $A > B + C$, una copia de seguridad optimizada ocupa menos espacio y se puede seleccionar. De lo contrario, si $A \leq B + C$, una copia de seguridad sin optimizar ocupa menos espacio y se puede seleccionar.

E. Realizaciones de restauración de optimización de datos a modo de ejemplo

Con referencia a la figura 13, se puede desear restaurar los flujos de datos optimizados a los cuales se les ha realizado una copia de seguridad en el almacenamiento de copia de seguridad 1304 a un almacenamiento primario en un cierto punto en el tiempo, tal como cuando el flujo de datos optimizado se ha corrompido de algún modo. Por ejemplo, uno o más fragmentos (por ejemplo, fragmento de asignación de flujo y/o fragmentos de datos) de un flujo de datos optimizado se pueden haber corrompido en el almacén de fragmentos 1334. En tal caso, se puede desear restaurar la versión a la que se ha realizado una copia de seguridad del flujo de datos optimizado a partir del almacenamiento de copia de seguridad 1304 en una forma sin optimizar. Es probable que las técnicas de copia de seguridad usadas en la actualidad realicen una copia de seguridad de espacios de nombres de sistema de archivos enteros, la cual incluye archivos optimizados y archivos de contenedor de almacén de fragmentos. Además, muchas técnicas de copia de seguridad usadas en la actualidad siguen usando medios de copia de seguridad secuenciales para el almacenamiento de copia de seguridad 2014, tal como medios de cinta. En ese sentido, es deseable que las técnicas para restaurar flujos de datos sean capaces de tener en cuenta diferentes tipos de medios de copia de seguridad y de formatos de copia de seguridad.

En esta subsección se describen realizaciones para restaurar flujos de datos optimizados a partir de un almacenamiento de copia de seguridad. Unas realizaciones posibilitan que se restauren flujos de datos optimizados (por ejemplo, archivos) de una forma que reduce la latencia y las operaciones de E/S de disco. De acuerdo con una realización, se habilita que los fragmentos para un flujo de datos particular se restauren a partir de un almacenamiento de copia de seguridad en lugar de restaurar porciones no necesarias adicionales de los contenedores de fragmentos con copia de seguridad. Unas realizaciones posibilitan una restauración selectiva de flujos de datos fuera de una copia de seguridad optimizada completa, sin requerir una comprensión de los metadatos de optimización de datos, y sin realizar suposiciones acerca del tipo de medios de copia de seguridad o el formato de copia de seguridad.

En unas realizaciones, la restauración de flujos de datos en sistemas de optimización de datos se puede realizar de varias formas. Por ejemplo, la figura 19 muestra un diagrama de flujo 1900 para realizar una restauración de datos de copia de seguridad en un sistema de optimización de datos, de acuerdo con una realización a modo de ejemplo. El diagrama de flujo 1900 puede ser realizado por la interfaz de almacén de fragmentos 116 de la figura 1, en una realización. El diagrama de flujo 1900 se describe con respecto a la figura 20 por razones de ilustración. La figura 20 muestra un diagrama de bloques de un sistema de restauración de datos 2000, de acuerdo con una realización a modo de ejemplo. Obsérvese que, en una realización, el sistema de copia de seguridad de datos 1300 de la figura 13 y el sistema de restauración de datos 2000 se pueden implementar en un sistema común (por ejemplo, en una

aplicación de copia de seguridad/restauración común, etc.) o se pueden implementar por separado. Tal como se muestra en la figura 20, el sistema de restauración de datos 2000 incluye un módulo de restauración de datos 2002 y un almacenamiento de copia de seguridad 2034. El almacenamiento de copia de seguridad 2034 almacena un contenedor de flujo 302 y uno o más contenedores de fragmentos 304 (por ejemplo, los contenedores de fragmentos 304a y 304b). Además, el módulo de restauración de datos 2002 incluye un reconstructor de archivos 2004 y una aplicación de restauración 2006. El diagrama de flujo 1900 puede ser realizado por el reconstructor de archivos 2004, en una realización. Realizaciones estructurales y operativas adicionales resultarán evidentes a los expertos en la materia o materias relevantes basándose en el análisis con respecto al diagrama de flujo 1900. El diagrama de flujo 1900 y el módulo de restauración de datos 2002 se describen tal como sigue.

Con referencia a la figura 20, la aplicación de restauración 2006 puede ser una aplicación de restauración convencional que está configurada para restaurar archivos a partir de un almacenamiento de copia de seguridad mediante la localización del archivo en un almacenamiento de copia de seguridad usando un catálogo (o cualquier otro mecanismo), y mediante la escritura del archivo localizado en un almacenamiento. La aplicación de restauración 2006 puede incluir solo una funcionalidad de restauración, o también puede incluir una funcionalidad de copia de seguridad (por ejemplo, puede ser una aplicación de copia de seguridad/restauración). En el caso de un sistema de optimización de datos, el archivo que se escribe en un almacenamiento por la aplicación de restauración 2006 en respuesta a una solicitud de restauración de archivo es la estructura de flujo optimizado, la cual contiene una referencia al fragmento de asignación de flujo.

Por ejemplo, la aplicación de restauración 2006 puede recibir una solicitud 2010 para que un flujo de datos se recupere del almacenamiento de copia de seguridad 2034. La solicitud 2010 puede identificar el flujo de datos por nombre de archivo. Tal como se muestra en la figura 20, la aplicación de restauración 2006 puede recuperar una estructura de flujo optimizado 2038 (por ejemplo, un código auxiliar de metadatos de flujo) a partir del almacenamiento de copia de seguridad 2034, que se corresponde con el flujo de archivo solicitado de la solicitud 2010. No obstante, debido a que la estructura de flujo optimizado 2038 es un punto de repetición de análisis sintáctico para el flujo de datos que se solicita en la solicitud 2010 (por ejemplo, la estructura de flujo optimizado 2038 no incluye datos de archivo), la aplicación de restauración 2006 está configurada para acceder al reconstructor de archivos 2004 para reconstruir el flujo de datos que se corresponde con la estructura de flujo optimizado 2038. Por ejemplo, el reconstructor de archivos 2004 se puede configurar para reconstruir el flujo de datos de acuerdo con el diagrama de flujo 1900 en la figura 19 tal como sigue.

En la etapa 1902 del diagrama de flujo 1900, se recibe una solicitud para que un flujo de datos se recupere de un almacén de fragmentos en almacenamiento, incluyendo la solicitud un identificador para metadatos de flujo optimizado que se corresponden con el flujo de datos. Por ejemplo, tal como se muestra en la figura 20, el reconstructor de archivos 2004 puede recibir la solicitud 2036 a partir de la aplicación de restauración 2006. La solicitud 2036 es una solicitud para que el reconstructor de archivos 2004 reconstruya el flujo de datos que se corresponde con la estructura de flujo optimizado 2038. La solicitud 2036 incluye la estructura de flujo optimizado 2038. La estructura de flujo optimizado 2038 incluye un indicador de metadatos de flujo optimizado, tal como un indicador de asignación de flujo, que se puede usar para localizar metadatos de flujo optimizado (por ejemplo, un fragmento de asignación de flujo) que se corresponden con el flujo de datos optimizado correspondiente. Tal como se describe en otra parte en el presente documento, los metadatos de flujo optimizado incluyen referencias a unos fragmentos de datos que pueden ser usados por el reconstructor de archivos 2004 para volver a ensamblar el flujo de datos.

El reconstructor de archivos 2004 se puede implementar en soporte físico, soporte lógico, soporte lógico inalterable o cualquier combinación de los mismos. Por ejemplo, en una realización, el reconstructor de archivos 2004 puede ser una interfaz de programación de aplicaciones (API, *application programming interface*) que está configurada para rehidratar flujos de datos basándose en una estructura de datos optimizada que se restaura a partir de un almacenamiento de copia de seguridad. En una realización, la API del reconstructor de archivos 2004 se puede configurar para recibir la solicitud 2036 en la siguiente forma:

Reconstruir Archivo ([en] estado de restauración/contexto, [en] nombre de archivo, [en] interfaz de devolución de llamada)

en la que los parámetros de solicitud se definen tal como sigue:

“estado de restauración/contexto” es una etiqueta que se usa para mantener el estado interno entre múltiples llamadas,

“nombre de archivo” es el nombre de archivo de la estructura de flujo optimizado 2038, e

“interfaz de devolución de llamada” identifica una interfaz de devolución de llamada para que el reconstructor de archivos 2004 la use para realizar llamadas de vuelta a la aplicación de restauración 2006.

En otras realizaciones, el reconstructor de archivos 2004 se puede configurar de unas formas que no sean las de una API. Por razones de ilustración, se puede hacer referencia a la realización de API para el reconstructor de archivos 2004 en lo sucesivo, pero la presente realización no tiene por objeto ser limitante.

En la etapa 1904, se genera una primera llamada a una aplicación de restauración basándose en el identificador de

metadatos de flujo optimizado, especificando la primera llamada un nombre de archivo para un primer contenedor de fragmentos en almacenamiento que almacena metadatos de flujo optimizado identificados por el identificador de metadatos de flujo optimizado, y especificando un desplazamiento para los metadatos de flujo optimizado en el primer contenedor de fragmentos. Tal como se muestra en la figura 20, en una realización, el reconstructor de archivos 2004 puede incluir un módulo de devolución de llamada 2008. El módulo de devolución de llamada 2008 está configurado para generar llamadas, y es usado por el reconstructor de archivos 2004 (por ejemplo, usado por la API del reconstructor de archivos 2004) para realizar una devolución de llamada a la aplicación de restauración 2006. El módulo de devolución de llamada 2008 puede realizar llamadas de vuelta a la aplicación de restauración 2006 (por ejemplo, de acuerdo con "interfaz de devolución de llamada" proporcionada por la aplicación de restauración 2006) para restaurar fragmentos a partir del almacenamiento de copia de seguridad 2034 que se identifican en las llamadas. El reconstructor de archivos 2004 recibe los fragmentos restaurados, y los usa para volver a ensamblar el flujo de datos identificado. Por ejemplo, en una realización, el módulo de devolución de llamada 2008 puede implementar llamadas de acuerdo con la siguiente estructura de devolución de llamada:

Mediante lectura ([ingresar en] estado de restauración/contexto, [ingresar en] nombre de archivo, [ingresar en] desplazamiento, [ingresar en] tamaño, [extraer de] memoria intermedia).
en la que los parámetros de llamada se definen tal como sigue:

"estado de restauración/contexto" es una etiqueta que se usa para mantener el estado interno entre múltiples llamadas,

"nombre de archivo" es un nombre de archivo para el contenedor en el almacenamiento de copia de seguridad 2034 que almacena el fragmento que se está solicitando,

"desplazamiento" es un desplazamiento en el contenedor para el fragmento que se está solicitando,

"tamaño" es un tamaño en el contenedor del fragmento que se está solicitando, y

"memoria intermedia" es una ubicación en la que la aplicación de restauración 2006 puede almacenar el fragmento restaurado para que el reconstructor de archivos 2004 acceda al fragmento restaurado (por ejemplo, una memoria intermedia en memoria u otro almacenamiento).

En otras realizaciones, unas llamadas pueden tener otros formatos o estructuras. La aplicación de restauración 2006 está configurada para una restauración selectiva al ser capaz de restaurar un fragmento a partir del almacenamiento de copia de seguridad 2034 a partir de un desplazamiento provisto en un archivo particular al que se le ha realizado una copia de seguridad. Por ejemplo, la aplicación de restauración 2006 puede usar un catálogo de copia de seguridad para asignar {nombre de archivo, desplazamiento de archivo} una llamada recibida a {medios de copia de seguridad, desplazamiento en medios} (para asignar archivos a medios de copia de seguridad).

En una realización en la que los metadatos de flujo optimizado se almacenan en forma de un fragmento de asignación de flujo, la primera llamada generada por el módulo de devolución de llamada 2008 (de acuerdo con la etapa 1904) es una solicitud a la aplicación de restauración 2006 para el fragmento de asignación de flujo que es identificado por el identificador de fragmento de asignación de flujo en la estructura de flujo optimizado 2038. En una realización, el módulo de devolución de llamada 2008 puede generar la primera llamada de acuerdo con la estructura de devolución de llamada que se ha descrito en lo que antecede. El módulo de devolución de llamada 2008 puede generar la primera llamada para incluir el valor del identificador de contenedor de fragmentos como el parámetro de "nombre de archivo", y para incluir el valor del valor de desplazamiento de fragmento como el parámetro de "desplazamiento". El parámetro de "tamaño" es opcional, debido a que la aplicación de restauración 2006 puede ser capaz de determinar de forma independiente el tamaño del fragmento solicitado en el almacenamiento de copia de seguridad 2034 (por ejemplo, debido a fragmentos de tamaño por defecto, al acceder a una información de tamaño de fragmento almacenado en un catálogo de copia de seguridad, etc.). Con respecto al ejemplo de la figura 20, el módulo de devolución de llamada 2008 puede generar la primera llamada para incluir el nombre de archivo del contenedor de flujo 302 como el parámetro de "nombre de archivo", y para incluir el valor del desplazamiento de la asignación de flujo en el contenedor de flujo 302 como el parámetro de "desplazamiento". Tal como se muestra en la figura 20, el reconstructor de archivos 2004 transmite la primera llamada 2014 generada por el módulo de devolución de llamada 2008 a la aplicación de restauración 2006.

En la etapa 1906, los metadatos de flujo optimizado se reciben en respuesta a la primera llamada. En una realización, la aplicación de restauración 2006 recibe la primera llamada 2014 y accede al almacenamiento de copia de seguridad 2034 para obtener el fragmento de asignación de flujo que se identifica en la primera llamada 2014. La aplicación de restauración 2006 genera un acceso 2016 al contenedor de flujo 302 en el almacenamiento de copia de seguridad 2034, y recupera un fragmento de asignación de flujo 2018 en el desplazamiento en el contenedor de flujo 302 que se indica en la primera llamada 2014. La aplicación de restauración 2006 transmite el fragmento de asignación de flujo 2018 al reconstructor de archivos 2004 en una respuesta 2020.

En la etapa 1908, se determina al menos un identificador de fragmento de datos al que se hace referencia en los metadatos de flujo optimizado. En una realización, el reconstructor de archivos 2004 analiza metadatos de la asignación de flujo (por ejemplo, los metadatos 314 de la asignación de flujo 310 de la figura 3) que son definidos por el fragmento de asignación de flujo 2018 que se recibe en la respuesta 2020 para uno o más identificadores de fragmento de datos para los fragmentos de datos a los que se hace referencia. Los metadatos de la asignación de flujo incluyen punteros en forma de identificadores de fragmento de datos (por ejemplo, el identificador de fragmento

de datos 404 de la figura 4) a cada uno de los fragmentos de datos 322 en los contenedores de fragmentos 304a y 304b que se incluyen en el flujo de datos que se está procesando. De esta forma, se determina o determinan los uno o más identificadores de fragmento de datos que se incluyen en el fragmento de asignación de flujo 2018.

En la etapa 1910, se genera al menos una llamada adicional a la aplicación de restauración en correspondencia con el al menos un identificador de fragmento de datos para obtener al menos un fragmento de datos a partir de al menos un contenedor de fragmentos en almacenamiento. En una realización, el módulo de devolución de llamada 2008 genera una o más llamadas adicionales a la aplicación de restauración 2006, con cada llamada siendo una solicitud para un fragmento de datos 322 en el almacenamiento de copia de seguridad 2034 que se corresponde con un identificador de fragmento de datos que se determina en la etapa 1908. Por ejemplo, el módulo de devolución de llamada 2008 puede generar una segunda llamada a la aplicación de restauración 2006 basándose en un identificador de fragmento de datos correspondiente del identificador o identificadores de fragmento de datos determinados. La segunda llamada puede tener la estructura de llamada que se ha descrito en lo que antecede, o puede tener otra estructura. La segunda llamada puede especificar un nombre de archivo (como el parámetro de "nombre de archivo") para uno de los contenedores de fragmentos 304a y 304b en el almacenamiento de copia de seguridad 2034 que almacena un fragmento de datos 322 que es identificado por el identificador de fragmento de datos. La segunda llamada puede especificar un desplazamiento (como el parámetro de "desplazamiento") para el fragmento de datos 322 identificado en el uno del primer y el segundo contenedores de fragmentos 304a y 304b que son identificados por el nombre de archivo. El parámetro de "tamaño" es opcional, debido a que la aplicación de restauración 2006 puede ser capaz de determinar de forma independiente el tamaño del fragmento de datos 322 identificado. Tal como se muestra en la figura 20, el reconstructor de archivos 2004 transmite la segunda llamada 2022 generada por el módulo de devolución de llamada 2008 a la aplicación de restauración 2006.

En la etapa 1912, el al menos un fragmento de datos se recibe en respuesta a la al menos una llamada adicional. En una realización, la aplicación de restauración 2006 recibe la segunda llamada 2022 y accede al almacenamiento de copia de seguridad 2034 para obtener el fragmento de datos 322 que se identifica en la segunda llamada 2022. Por ejemplo, si el fragmento de datos identificado se almacena en el primer contenedor de fragmentos 304a, la aplicación de restauración 2006 puede generar un acceso 2024 al primer contenedor de fragmentos 302a en el almacenamiento de copia de seguridad 2034, y puede recuperar un fragmento de datos 2026 en el desplazamiento en el primer contenedor de fragmentos 302a que se indica en la segunda llamada 2022. La aplicación de restauración 2006 transmite el fragmento de datos 2026 al reconstructor de archivos 2004 en una respuesta 2028.

Obsérvese que, para cada identificador de fragmento de datos adicional que se determina en la etapa 1908, el constructor de archivos 2004 puede generar una llamada adicional a la aplicación de restauración 2006 de una forma similar a la de la segunda llamada 2022 para recuperar el fragmento de datos 322 correspondiente a partir del almacenamiento de copia de seguridad 2034. Por ejemplo, en respuesta a una tercera llamada que se recibe del reconstructor de archivos 2004, la aplicación de restauración 2006 puede acceder al almacenamiento de copia de seguridad 2034 para obtener el fragmento de datos 322 que se identifica en la tercera llamada. Por ejemplo, si el fragmento de datos identificado se almacena en el segundo contenedor de fragmentos 304b, la aplicación de restauración 2006 puede generar un acceso 2030 al segundo contenedor de fragmentos 302b en el almacenamiento de copia de seguridad 2034, y puede recuperar un fragmento de datos 2032 en el desplazamiento en el segundo contenedor de fragmentos 302b que se indica en la tercera llamada. La aplicación de restauración 2006 puede transmitir el fragmento de datos 2032 al reconstructor de archivos 2004 en otra respuesta. Llamadas adicionales que se corresponden con identificadores de fragmento de datos adicionales se pueden procesar de una forma similar para recuperar fragmentos de datos adicionales a partir del almacenamiento de copia de seguridad 2034 para el reconstructor de archivos 2004.

En la etapa 1914, los metadatos de flujo optimizado y el al menos un fragmento de datos se combinan para restaurar el flujo de datos. En una realización, el fragmento de asignación de flujo 2018 y los uno o más fragmentos de datos 322 (por ejemplo, el fragmento de datos 2026 y 2032) que se recuperan del almacenamiento de copia de seguridad 2034 son combinables para restaurar el flujo de datos. De una forma similar a como se ha descrito en lo que antecede para el módulo de rehidratación 702, el reconstructor de archivos 2004 puede usar los desplazamientos de flujo de datos 402 (la figura 4) que se incluyen en la asignación de flujo del fragmento de asignación de flujo 2018 para disponer los fragmentos de datos 322 recuperados en el orden apropiado para regenerar el flujo de datos, el cual es emitido por el reconstructor de archivos 2004 como el flujo de datos 2012. El flujo de datos 2012 se puede almacenar en un almacenamiento primario, se puede emitir a la aplicación de restauración 2006 para almacenarse en un almacenamiento primario o se puede manejar o entregar de otro modo, según se desee para la aplicación particular.

Por razones de ilustración, una restauración a modo de ejemplo de un archivo optimizado se describe con referencia al diagrama de flujo 1900 de la figura 19 y el sistema 2000 de la figura 20. El fragmento de asignación de flujo para el archivo optimizado se almacena en el contenedor de flujo 302, y el archivo optimizado incluye los fragmentos de datos 322a y 322b del contenedor de fragmentos 304a y el fragmento de datos 322e del contenedor de fragmentos 304b. En el presente ejemplo, el contenedor de flujo 302 tiene el nombre de archivo SC5, el contenedor de fragmentos 304a tiene el nombre de archivo CC3, y el contenedor de fragmentos 304b tiene el nombre de archivo CC7. Tal como se ha descrito en lo que antecede, la aplicación de restauración 2006 puede restaurar la estructura de flujo optimizado 2038 (que tiene el nombre de archivo "OFN", por ejemplo) para el archivo optimizado a partir del

almacenamiento de copia de seguridad 2034. En la etapa 1902, la aplicación de restauración 2006 puede generar la solicitud 2036 al reconstructor de archivos 2004 (por ejemplo, usando Reconstruir Archivo (OFN, Devolución de Llamada)) para comenzar el procedimiento de reconstrucción de archivos para el archivo optimizado. En la etapa 1904, el módulo de devolución de llamada 2008 puede generar una primera llamada a la aplicación de restauración 2006 solicitando al fragmento de asignación de flujo que se identifica en la estructura de flujo optimizado 2038 el archivo optimizado (por ejemplo, Devolución de Llamada -> Leer (estado interno, SC5, desplazamiento de fragmento de asignación de flujo, 64 K, memoria intermedia), en la que un tamaño del fragmento de asignación de flujo es 64 K). En la etapa 1906, la aplicación de restauración 2006 recupera el fragmento de asignación de flujo a partir del contenedor de flujo 304, y proporciona el fragmento de asignación de flujo al reconstructor de archivos 2004 en una respuesta. En la etapa 1908, el reconstructor de archivos 2004 analiza el fragmento de asignación de flujo para determinar los tres identificadores de fragmento de datos para los fragmentos de datos 322a, 322b y 322e que se incluyen en el archivo deseado. En la etapa 1910, el módulo de devolución de llamada 2008 genera tres llamadas a la aplicación de restauración 2006 solicitando los fragmentos de datos 322a, 322b y 322e, de forma respectiva (por ejemplo, una primera llamada de Devolución de Llamada -> Leer (estado interno, CC3, desplazamiento del fragmento de datos 322a, 64 K, memoria intermedia), una segunda llamada de Devolución de Llamada -> Leer (estado interno, CC3, desplazamiento del fragmento de datos 322b, 64 K, memoria intermedia), y una tercera llamada de Devolución de Llamada -> Leer (estado interno, CC7, desplazamiento del fragmento de datos 322e, 64 K, memoria intermedia)). En la etapa 1912, el reconstructor de archivos 2004 recibe los fragmentos de datos 322a, 322b y 322e a partir de la aplicación de restauración 2006 en respuestas a las tres llamadas. En la etapa 1914, el reconstructor de archivos 2004 vuelve a ensamblar el archivo optimizado en una forma no optimizada de acuerdo con la asignación de flujo del fragmento de asignación de flujo y almacena el archivo ensamblado (por ejemplo, escribe el archivo vuelto a ensamblar de vuelta en la estructura de flujo optimizado 2038 en el disco).

Se hace notar que, en el ejemplo del diagrama de flujo 1900, el flujo de datos se vuelve a ensamblar (en la etapa 1914). En otra realización, en lugar del nuevo ensamblado del flujo de datos, la estructura de flujo optimizado 2038 se puede mantener en un almacenamiento (primario o de copia de seguridad) como un archivo optimizado, y los fragmentos de datos que se determina que están incluidos en el flujo de datos se pueden almacenar en un almacén de fragmentos (no en el almacenamiento de copia de seguridad 2034). De esta forma, el flujo de datos optimizado permanece en una forma optimizada, haciendo referencia a los fragmentos de datos en un almacén de fragmentos. En una realización de este tipo, el fragmento de asignación de flujo (que se recibe en la etapa 1906) y los uno o más fragmentos de datos (que se reciben en la etapa 1912) se almacenan en el almacén de fragmentos.

Tal como se ha descrito adicionalmente en lo que antecede, los fragmentos pueden ser movidos dentro de contenedores debido a varias funciones, tal como desfragmentación y compactación. En ese sentido, se puede mantener una tabla de redirección que realiza un seguimiento de tal movimiento de fragmentos. En una realización, el reconstructor de archivos 2004 se puede configurar para acceder a una tabla de redirección para ajustar llamadas para tener fragmentos recuperados de desplazamientos ajustados en un contenedor. Por ejemplo, la figura 21 muestra un diagrama de bloques de módulo de devolución de llamada que accede a la tabla de redistribución 900, de acuerdo con una realización. El módulo de devolución de llamada 2008 puede acceder a una tabla de redistribución 900 para cada contenedor en un almacenamiento de copia de seguridad (por ejemplo, cada uno del contenedor de flujo 302 y los contenedores de fragmentos 304a y 304b en el ejemplo de la figura 20). De esta forma, el módulo de devolución de llamada 2008 puede generar llamadas en desplazamientos ajustados en contenedores en lugar de en desplazamientos obsoletos.

Por ejemplo, en una realización, el módulo de devolución de llamada 2008 puede realizar un diagrama de flujo 2200 que se muestra en la figura 22. La figura 22 muestra el diagrama de flujo 2200 que proporciona un procedimiento para acceder a una tabla de redistribución para obtener desplazamientos actualizados para los fragmentos, de acuerdo con una realización a modo de ejemplo. El diagrama de flujo 2200 se describe tal como sigue.

En la etapa 2202 del diagrama de flujo 2200, se accede a una tabla de redirección para asignar un primer desplazamiento que se incluye en un identificador de fragmento de datos para un fragmento de datos a un segundo desplazamiento. Por ejemplo, el módulo de devolución de llamada 2008 puede acceder a una tabla de redistribución de un contenedor de flujo a partir de la cual se va a recuperar un fragmento de asignación de flujo (por ejemplo, la tabla de redirección 308 en el contenedor de flujo 302 de la figura 3) o una redistribución de un contenedor de fragmentos a partir de la cual se va a recuperar un fragmento de datos (por ejemplo, la tabla de redistribución 320 en el contenedor de fragmentos 304 de la figura 3). El módulo de devolución de llamada 2008 puede acceder a la tabla de redistribución de cualquier forma, incluyendo mediante la generación de una llamada para recuperar la tabla de redistribución. Por ejemplo, una llamada se puede generar y transmitir a la aplicación de restauración 2006 que define un nombre de archivo del contenedor que contiene la tabla de redistribución, y un desplazamiento para la tabla de redistribución en el contenedor. La aplicación de restauración 2006 puede recuperar la tabla de redistribución a partir del almacenamiento de copia de seguridad 2034, y puede proporcionar la tabla de redistribución al reconstructor de archivos 2004. De esta forma, el módulo de devolución de llamada 2008 puede acceder a la tabla de redistribución para determinar desplazamientos de fragmento actualizados.

En la etapa 2204, se genera una segunda llamada a la aplicación de restauración basándose en el identificador de fragmento de datos, especificando la segunda llamada un nombre de archivo para un segundo contenedor de fragmentos en almacenamiento que almacena un fragmento de datos que se corresponde con el identificador de

fragmento de datos, y especificando el segundo desplazamiento para el fragmento de datos en el segundo contenedor de fragmentos. Por ejemplo, antes de una llamada para un fragmento de datos, el módulo de devolución de llamada 2008 puede acceder a la tabla de redistribución para el fragmento de datos para determinar un segundo desplazamiento actualizado para el fragmento de datos (si un nuevo desplazamiento se encuentra presente en la tabla de redistribución). El módulo de devolución de llamada 2008 puede usar el valor de desplazamiento actualizado en una llamada a la aplicación de restauración 2006 para recuperar el fragmento de datos. La llamada puede especificar el nombre de archivo del contenedor de fragmentos 304 en el almacenamiento de copia de seguridad 2034, y puede especificar el valor de desplazamiento actualizado que se obtiene de la tabla de redistribución. La aplicación de restauración 2006 puede recibir la llamada, recuperar el fragmento de datos a partir del contenedor de fragmentos indicado y el desplazamiento actualizado y proporcionar el fragmento de datos recuperado al reconstructor de archivos 2004 en una respuesta.

Obsérvese que, en el ejemplo del diagrama de flujo 2200, se accede a una tabla de redistribución de un contenedor de fragmentos para asignar un desplazamiento de fragmento de datos de un primer desplazamiento a un segundo desplazamiento. Como alternativa, el diagrama de flujo 2200 se puede modificar de tal modo que se accede a una tabla de redistribución de un contenedor de flujo para asignar un fragmento de asignación de flujo de un primer desplazamiento a un segundo desplazamiento. En una realización, las tablas de redirección de contenedores pueden ser recuperadas por adelantado por el módulo de devolución de llamada 2008, antes de generar llamadas para los fragmentos.

En una realización, un tamaño de los fragmentos que se están recuperando del almacenamiento de copia de seguridad 2034 se determina antes de la recuperación de los fragmentos. Por ejemplo, en una realización, para cada fragmento, una primera llamada puede ser hecha por el módulo de devolución de llamada 2008 a la aplicación de restauración 2006 para determinar un tamaño de un fragmento que se va a recuperar, y una segunda llamada puede ser hecha por el módulo de devolución de llamada 2008 a la aplicación de restauración 2006 para recuperar el fragmento a partir del almacenamiento de copia de seguridad 2034 que tiene el tamaño determinado.

Por ejemplo, en una realización, el módulo de devolución de llamada 2008 puede generar una primera llamada que especifica el nombre de archivo para el contenedor en el almacenamiento de copia de seguridad 2034 que almacena el fragmento, un desplazamiento para el fragmento en el contenedor, y un tamaño de encabezado para el fragmento (y, opcionalmente, una memoria intermedia en la cual emitir el fragmento). Por ejemplo, el tamaño del encabezado de fragmento puede ser un tamaño de encabezado convencional que es conocido por el módulo de devolución de llamada 2008. La primera llamada se puede transmitir a la aplicación de restauración 2006 para recuperar el encabezado de fragmento a partir del almacenamiento de copia de seguridad 2034 tal como se indica por la primera llamada. La aplicación de restauración 2006 puede recuperar el encabezado de fragmento (por ejemplo, o bien un encabezado de fragmento de asignación de flujo o bien un encabezado de fragmento de datos), y puede proporcionar el encabezado de fragmento al módulo de devolución de llamada 2008 en una primera respuesta. El encabezado de fragmento puede indicar un tamaño del fragmento. Por ejemplo, un encabezado de fragmento de asignación de flujo puede indicar un tamaño del fragmento de asignación de flujo, y un encabezado de fragmento de datos puede indicar un tamaño del fragmento de datos. El módulo de devolución de llamada 2008 puede generar entonces una segunda llamada que especifica el nombre de archivo para el contenedor en el almacenamiento de copia de seguridad 2034 que almacena el fragmento, un desplazamiento para el fragmento en el contenedor, y el tamaño para el fragmento que se determina a partir del encabezado de fragmento (y, opcionalmente, una memoria intermedia en la cual emitir el fragmento). La aplicación de restauración 2006 puede recuperar el fragmento (por ejemplo, o bien un fragmento de asignación de flujo o bien un fragmento de datos) con el desplazamiento indicado, y teniendo el tamaño indicado. La aplicación de restauración 2006 puede proporcionar el fragmento recuperado al módulo de devolución de llamada 2008 en una segunda respuesta. De esta forma, el módulo de devolución de llamada 2008 puede usar la aplicación de restauración 2006 para determinar tamaños de fragmento, y se pueden usar dos llamadas para recuperar fragmentos: se puede usar una primera llamada para determinar el tamaño de fragmento, y se puede usar una segunda llamada para recuperar el fragmento que tiene el tamaño determinado.

En una realización, el módulo de devolución de llamada 2008 puede generar llamadas para recuperar fragmentos de datos en cualquier orden, incluyendo en un orden en el que los identificadores de fragmento de datos se incluyen en el fragmento de asignación de flujo. En otra realización, el reconstructor de archivos 2004 se puede configurar para clasificar identificadores de fragmento de datos para que el módulo de devolución de llamada 2008 recupere los fragmentos de datos en un orden más eficiente. Por ejemplo, la figura 23 muestra un diagrama de bloques del reconstructor de archivos 2004, de acuerdo con una realización a modo de ejemplo. Tal como se muestra en la figura 23, el reconstructor de archivos 2004 incluye un clasificador de identificadores 2302. El clasificador de identificadores 2302 está configurado para clasificar un orden en el que se restauran los fragmentos de datos. Por ejemplo, en una realización, el clasificador de identificadores 2302 puede realizar un diagrama de flujo 2400 que se muestra en la figura 24. La figura 24 muestra un diagrama de flujo 2400 que proporciona un procedimiento para clasificar un orden de restauración de fragmento de datos, de acuerdo con una realización a modo de ejemplo. El diagrama de flujo 2400 se describe tal como sigue.

En la etapa 2402 del diagrama de flujo 2400, la pluralidad de identificadores de fragmento de datos se clasifican basándose en un contenedor de fragmentos determinado como una primera clave de clasificación y un desplazamiento de fragmento como una segunda clave. En una realización, el clasificador de identificadores 2302

puede clasificar los identificadores de fragmento de datos que se incluyen en un fragmento de asignación de flujo de acuerdo con una o más claves de clasificación para posibilitar que los fragmentos de datos correspondientes se recuperen de forma más eficiente. Por ejemplo, los identificadores de fragmento de datos se pueden clasificar basándose en archivo de contenedor como una primera clave de clasificación y basándose en un fragmento dentro del archivo de contenedor como una segunda clave de clasificación. En unas realizaciones adicionales, los identificadores de fragmento de datos se pueden clasificar usando criterios o claves adicionales y/o alternativos.

En la etapa 2404, se genera una pluralidad de llamadas adicionales a la aplicación de restauración en correspondencia con la pluralidad de identificadores de fragmento de datos en un orden que es definido por la etapa 2402. Por ejemplo, las llamadas adicionales de la etapa 1910 del diagrama de flujo 1900 (la figura 19) se pueden generar en un orden que es definido por la clasificación de la etapa 2402. De esta forma, los fragmentos de datos se pueden recuperar de forma más eficiente del almacenamiento de copia de seguridad 2034. Por ejemplo, las llamadas se pueden clasificar de tal modo que los fragmentos de datos se recuperan en un orden secuencial que coincide con una secuencia en la que los fragmentos de datos se almacenan en un medio secuencial, tal como cinta, y de tal modo que los fragmentos de datos que se almacenan en un medio de copia de seguridad común (por ejemplo, una misma cinta, un mismo disco, etc.) se restauran de forma secuencial antes de restaurar otros fragmentos de datos. Tal ordenación puede reducir un número de conmutación de medios de copia de seguridad y de búsquedas dentro de los medios de copia de seguridad, lo que es particularmente deseable para una restauración a partir de cinta.

El rendimiento de las realizaciones de restauración se puede mejorar si la aplicación de restauración puede tener influencia en el orden de restauración y/o recibir un aviso anticipado de qué extensiones de archivo están a punto de restaurarse a partir de cada archivo de contenedor. Por ejemplo, una o más de las siguientes funciones de devolución de llamada pueden ser usadas por el reconstructor de archivos para mejorar la restauración, en unas realizaciones:

“ContainersRestoreOrder” es una devolución de llamada de orden de restauración que recibe una matriz con todos los nombres de archivo de contenedor de fragmentos de datos como una entrada. Los nombres de archivo son de aquellos contenedores que son necesarios para la reconstrucción de archivos (los contenedores que almacenan al menos un fragmento de datos que se requiere para la reconstrucción de archivos). Una salida de esta función de devolución de llamada es una matriz con la misma lista de nombres de archivo de contenedor, con la lista clasificada (por ejemplo, reordenada).

Usando esta devolución de llamada de orden de restauración, en el caso de una copia de seguridad en cinta, la aplicación de restauración puede clasificar los nombres de archivo de contenedor basándose en orden de cinta, de acuerdo con las cintas de copia de seguridad y un desplazamiento en la cinta en la que se almacena el archivo de contenedor. Si el procedimiento de restauración se realiza de acuerdo con este orden de clasificación, las funciones de sustitución de medios de cinta (por ejemplo, por un usuario) y de búsqueda de cinta se pueden reducir o reducir al mínimo. Reducir al mínimo el número de cambios de medios de cinta y de búsquedas de cinta aumenta la vida útil de los medios de cinta y acelera el procedimiento de restauración. En el caso en el que las cintas que contienen estos archivos de contenedor están ubicadas físicamente en otro sitio y es necesario que se soliciten y se transporten desde una ubicación fuera del sitio, esta capacidad posibilita que la aplicación de restauración proporcione una lista de cintas que es necesario que se recuperen de la ubicación fuera del sitio para completar la restauración de los datos. Esta capacidad puede acelerar de forma significativa la recuperación de datos debido a que la recuperación de cinta fuera del sitio puede llevar horas o incluso días, y puede reducir el coste asociado con la recuperación de medios.

Por ejemplo, la aplicación de restauración puede en primer lugar agrupar los archivos de contenedor basándose en en qué cinta de copia de seguridad se encuentran estos, y dentro de cada cinta, los archivos de contenedor se pueden clasificar basándose en su ubicación (desplazamiento de partida) en la cinta. El reconstructor de archivos puede restaurar entonces los fragmentos de datos basándose en el orden de contenedores que es especificado por la aplicación de copia de seguridad. La aplicación de copia de seguridad puede llamar a la función de devolución de llamada leída para los fragmentos de datos que se almacenan en el primer archivo de contenedor de fragmentos de datos en la lista clasificada, puede llamar a continuación a la devolución de llamada leída para los fragmentos de datos que se almacenan en el segundo archivo de contenedor de fragmentos de datos en la lista clasificada, y así sucesivamente.

En el caso de copias de seguridad de archivos u otras técnicas de copia de seguridad que no se benefician de la ordenación de la restauración, la aplicación de restauración puede optar por evitar clasificar la lista de contenedores de fragmentos de datos y dejar el orden de restauración de contenedores arbitrario.

“ContainerRestorePlan” es una devolución de llamada de plan de restauración de contenedor que recibe un único nombre de archivo de contenedor de fragmentos de datos y una matriz de desplazamientos y tamaños de fragmentos de datos como entrada. La matriz incluye todos los fragmentos de datos que están almacenados en el contenedor nombrado y que son necesarios para la reconstrucción del archivo.

El reconstructor de archivos puede llamar a esta devolución de llamada una vez por cada contenedor de fragmentos

de datos, antes de que se llame a la devolución de llamada leída para los fragmentos de datos que se almacenan en el contenedor de fragmentos de datos. La devolución de llamada `ContainerRestorePlan` genera un plan de restauración para su uso por la aplicación de restauración. La aplicación de restauración puede usar el plan de restauración que es emitido por la devolución de llamada, el cual indica las extensiones de archivo del archivo de contenedor de fragmentos de datos que es necesario leer a partir de los medios de copia de seguridad. La aplicación de restauración puede usar este plan de restauración para su propia optimización de captura de los fragmentos de datos a partir de los medios de copia de seguridad, usando unas técnicas tales como procesamiento por lotes, almacenamiento en memoria caché y lectura anticipada.

El procesamiento de esta devolución de llamada es opcional. Si la aplicación de restauración no puede o no necesita optimizar las lecturas a partir de los medios de copia de seguridad, entonces la aplicación de restauración puede ignorar el plan y procesar las devoluciones de llamada leídas sin optimizaciones.

“`CreateChunksRestorePlan`” es una devolución de llamada de plan de restauración de fragmento de datos que recibe una matriz de estructuras como entrada. Cada estructura en la matriz se corresponde con un único fragmento de datos que es necesario para reconstruir el archivo. Cada estructura puede incluir los siguientes miembros: nombre de archivo de contenedor (que indica en dónde se almacena el fragmento de datos), desplazamiento de fragmento en el archivo de contenedor y tamaño de fragmento. La salida de la devolución de llamada es una matriz de las mismas estructuras, que se clasifican en el orden en el que se va a realizar la restauración.

La devolución de llamada de plan de restauración de fragmento de datos es una alternativa a la devolución de llamada de orden de restauración y la devolución de llamada de plan de restauración de contenedor. La devolución de llamada de plan de restauración de fragmento de datos sirve para al menos dos fines: en primer lugar, la devolución de llamada indica a la aplicación de restauración qué fragmentos de datos se van a restaurar. Basándose en la entrada a la devolución de llamada, la aplicación de restauración puede determinar qué medios de copia de seguridad tiene cada fragmento, y en dónde está ubicado cada fragmento en los medios correspondientes. Se pueden usar las mismas técnicas de optimización que para leer datos de medios de copia de seguridad. En segundo lugar, basándose en la salida de la devolución de llamada, la aplicación de restauración puede indicar al reconstructor de archivos en qué orden restaurar los fragmentos.

Por ejemplo, en el caso de una copia de seguridad en cinta, es posible que un archivo de contenedor de fragmentos de datos particular se almacene en múltiples cintas. La reconstrucción de un archivo puede requerir dos fragmentos de datos que están ubicados en el mismo archivo de contenedor, pero de los cuales se realiza una copia de seguridad en 2 cintas diferentes, debido a que el archivo de contenedor está fraccionado entre las cintas. Si la aplicación de restauración tiene la oportunidad de dictar el orden de restauración a la granularidad de los datos de fragmento, la aplicación de restauración puede agrupar y clasificar fragmentos de datos en orden de cinta, incluso para aquellos casos en los que el archivo de contenedor está fraccionado entre múltiples cintas.

El reconstructor de archivos puede llamar a la devolución de llamada de plan de restauración de fragmento de datos una vez por cada procedimiento de restauración del archivo. El reconstructor de archivos puede llamar a la devolución de llamada en el punto en el que este conoce qué fragmentos de datos son necesarios, y en dónde se encuentran ubicados los fragmentos de datos en los archivos de contenedor de fragmentos de datos. El reconstructor de archivos puede usar la salida de la devolución de llamada para proporcionar el orden de la restauración de fragmento de datos. El reconstructor de archivos puede llamar a la función de devolución de llamada leída basándose en el orden de los fragmentos de datos que dicta la lista de estructuras de salida.

El procesamiento de la devolución de llamada de plan de restauración de fragmento de datos es opcional. Si la aplicación de restauración no puede o no necesita optimizar las lecturas de medios de copia de seguridad y dictar el orden de restauración, entonces la devolución de llamada se puede ignorar o no realizarse. El reconstructor de archivos puede restaurar entonces los fragmentos de datos usando su propia lógica que se ha descrito anteriormente, o en un cierto orden arbitrario.

III. Realizaciones de dispositivos informáticos a modo de ejemplo

El módulo de deduplicación de datos 104, el módulo de mantenimiento 106, la API de flujo de datos 110, la API de mantenimiento de fragmentos 112, la API de acceso de datos 114, la interfaz de almacén de fragmentos 116, la unidad de análisis sintáctico de flujo de datos 402, el gestor de almacenamiento de fragmentos de datos 404, el generador de metadatos 406, el generador de asignaciones de flujo 408, el módulo de rehidratación 702, el ensamblador de flujos de datos 1102, la unidad de verificación de generación 1106, la unidad de recuperación de fragmentos de datos 1108, el módulo de copia de seguridad de datos 1302, el módulo de copia de seguridad de archivos optimizada 1306, el módulo de copia de seguridad de rehidratación 1308, el módulo de copia de seguridad a nivel de elemento 1310, el módulo de copia de seguridad de identificadores de fragmento de datos 1312, el módulo de heurística 1314, el módulo de restauración de datos 2002, el reconstructor de archivos 2004, la aplicación de restauración 2006, el módulo de devolución de llamada 2008 y el clasificador de identificadores 2302 se pueden implementar en soporte físico, soporte lógico, soporte lógico inalterable o cualquier combinación de los mismos. Por ejemplo, el módulo de deduplicación de datos 104, el módulo de mantenimiento 106, la API de flujo de datos 110, la API de mantenimiento de fragmentos 112, la API de acceso de datos 114, la interfaz de almacén de fragmentos 116,

la unidad de análisis sintáctico de flujo de datos 402, el gestor de almacenamiento de fragmentos de datos 404, el generador de metadatos 406, el generador de asignaciones de flujo 408, el módulo de rehidratación 702, el ensamblador de flujos de datos 1102, la unidad de verificación de generación 1106, la unidad de recuperación de fragmentos de datos 1108, el módulo de copia de seguridad de datos 1302, el módulo de copia de seguridad de archivos optimizada 1306, el módulo de copia de seguridad de rehidratación 1308, el módulo de copia de seguridad a nivel de elemento 1310, el módulo de copia de seguridad de identificadores de fragmento de datos 1312, el módulo de heurística 1314, el módulo de restauración de datos 2002, el reconstructor de archivos 2004, la aplicación de restauración 2006, el módulo de devolución de llamada 2008 y/o el clasificador de identificadores 2302 se pueden implementar como un código de programa informático que está configurado para ejecutarse en uno o más procesadores. Como alternativa, el módulo de deduplicación de datos 104, el módulo de mantenimiento 106, la API de flujo de datos 110, la API de mantenimiento de fragmentos 112, la API de acceso de datos 114, la interfaz de almacén de fragmentos 116, la unidad de análisis sintáctico de flujo de datos 402, el gestor de almacenamiento de fragmentos de datos 404, el generador de metadatos 406, el generador de asignaciones de flujo 408, el módulo de rehidratación 702, el ensamblador de flujos de datos 1102, la unidad de verificación de generación 1106, la unidad de recuperación de fragmentos de datos 1108, el módulo de copia de seguridad de datos 1302, el módulo de copia de seguridad de archivos optimizada 1306, el módulo de copia de seguridad de rehidratación 1308, el módulo de copia de seguridad a nivel de elemento 1310, el módulo de copia de seguridad de identificadores de fragmento de datos 1312, el módulo de heurística 1314, el módulo de restauración de datos 2002, el reconstructor de archivos 2004, la aplicación de restauración 2006, el módulo de devolución de llamada 2008 y/o el clasificador de identificadores 2302 se pueden implementar como lógica/conjunto de circuitos eléctricos de soporte físico.

La figura 25 ilustra una implementación a modo de ejemplo de un ordenador 2500 en la que se pueden implementar realizaciones de la presente invención. Por ejemplo, el sistema de almacenamiento 102, y/o cualquier porción del mismo, se pueden implementar en uno o más sistemas informáticos similares al ordenador 2500, incluyendo uno o más rasgos distintivos del ordenador 2500 y/o rasgos distintivos alternativos. El ordenador 2500 puede ser un dispositivo informático de propósito general en forma de un ordenador personal convencional, un equipo informático portátil o una estación de trabajo, por ejemplo, o el ordenador 2500 puede ser un dispositivo informático de propósito especial. La descripción del ordenador 2500 que se proporciona en el presente documento se proporciona por razones de ilustración, y no tiene por objeto ser limitante. Se pueden implementar realizaciones de la presente invención en tipos adicionales de sistemas informáticos, tal como sería conocido por los expertos en la materia o materias relevantes.

Tal como se muestra en la figura 25, el ordenador 2500 incluye una unidad de procesamiento 2502, una memoria de sistema 2504 y un bus 2506 que acopla diversos componentes de sistema que incluyen la memoria de sistema 2504 con la unidad de procesamiento 2502. El bus 2506 representa uno o más de cualquiera de varios tipos de estructuras de bus, incluyendo un bus de memoria o un controlador de memoria, un bus de periféricos, un puerto de gráficos acelerados y un bus local o de procesador usando cualquiera de una diversidad de arquitecturas de bus. La memoria de sistema 2504 incluye una memoria de solo lectura (ROM, *read only memory*) 2508 y una memoria de acceso aleatorio (RAM, *random access memory*) 2510. Un sistema básico de entrada/salida 2512 (BIOS, *basic input/output system*) se almacena en la ROM 2508.

El ordenador 2500 también tiene una o más de las siguientes unidades: una unidad de disco duro 2514 para leer de y escribir en un disco duro, una unidad de disco magnético 2516 para leer de o escribir en un disco magnético extraíble 2518, y una unidad de disco óptico 2520 para leer de o escribir en un disco óptico extraíble 2522 tal como un CD ROM, un DVD ROM, u otros medios ópticos. La unidad de disco duro 2514, la unidad de disco magnético 2516 y la unidad de disco óptico 2520 están conectadas con el bus 2506 mediante una interfaz de unidad de disco duro 2524, una interfaz de unidad de disco magnético 2526 y una interfaz de unidad óptica 2528, de forma respectiva. Las unidades y sus medios legibles por ordenador asociados proporcionan un almacenamiento no volátil de instrucciones legibles por ordenador, estructuras de datos, módulos de programa y otros datos para el ordenador. A pesar de que se describen un disco duro, un disco magnético extraíble y un disco óptico extraíble, se pueden usar otros tipos de medios de almacenamiento legibles por ordenador para almacenar datos, tales como tarjetas de memoria flash, discos de vídeo digital, memorias de acceso aleatorio (RAM, *random access memory*), memorias de solo lectura (ROM, *read only memory*), y similares.

Un número de módulos de programa se pueden almacenar en el disco duro, el disco magnético, el disco óptico, la ROM o la RAM. Estos programas incluyen un sistema operativo 2530, uno o más programas de aplicación 2532, otros módulos de programa 2534 y unos datos de programa 2536. Los programas de aplicación 2532 o los módulos de programa 2534 pueden incluir, por ejemplo, una lógica de programa informático para implementar el módulo de deduplicación de datos 104, el módulo de mantenimiento 106, la API de flujo de datos 110, la API de mantenimiento de fragmentos 112, la API de acceso de datos 114, la interfaz de almacén de fragmentos 116, la unidad de análisis sintáctico de flujo de datos 402, el gestor de almacenamiento de fragmentos de datos 404, el generador de metadatos 406, el generador de asignaciones de flujo 408, el módulo de rehidratación 702, el ensamblador de flujos de datos 1102, la unidad de verificación de generación 1106, la unidad de recuperación de fragmentos de datos 1108, el módulo de copia de seguridad de datos 1302, el módulo de copia de seguridad de archivos optimizada 1306, el módulo de copia de seguridad de rehidratación 1308, el módulo de copia de seguridad a nivel de elemento 1310, el módulo de copia de seguridad de identificadores de fragmento de datos 1312, el módulo de heurística 1314, el módulo de restauración de datos 2002, el reconstructor de archivos 2004, la aplicación

de restauración 2006, el módulo de devolución de llamada 2008, el clasificador de identificadores 2302, el diagrama de flujo 500, el diagrama de flujo 1000, el diagrama de flujo 1200, el diagrama de flujo 1400, el diagrama de flujo 1500, el diagrama de flujo 1600, el diagrama de flujo 1700, el diagrama de flujo 1800, el diagrama de flujo 1900, el diagrama de flujo 2200, el diagrama de flujo 2400 (incluyendo cualquier etapa de los diagramas de flujo 500, 1000, 1200, 1400, 1500, 1600, 1700, 1800, 1900, 2200 y 2400), y/o realizaciones adicionales que se describen en el presente documento.

Un usuario puede introducir instrucciones e información en el ordenador 2500 a través de unos dispositivos de entrada tales como el teclado 2538 y el dispositivo señalador 2540. Otros dispositivos de entrada (que no se muestran) pueden incluir un micrófono, una palanca de mando, un controlador para juegos, una antena parabólica, un escáner, o similares. Estos y otros dispositivos de entrada a menudo están conectados con la unidad de procesamiento 2502 a través de una interfaz de puerto serie 2542 que está acoplada con el bus 2506, pero se pueden conectar mediante otras interfaces, tal como un puerto paralelo, puerto de juegos, o un bus serie universal (USB, *universal serial bus*).

Un dispositivo de presentación visual 2544 también está conectado con el bus 2506 por medio de una interfaz, tal como un adaptador de vídeo 2546. Además del monitor, el ordenador 2500 puede incluir otros dispositivos de salida periféricos (que no se muestran) tales como altavoces e impresoras.

El ordenador 2500 está conectado con una red 2548 (por ejemplo, Internet) a través de un adaptador o interfaz de red 2550, un módem 2552, u otros medios para establecer comunicaciones a través de la red. El módem 2552, el cual puede ser interno o externo, está conectado con el bus 2506 por medio de la interfaz de puerto serie 2542.

Tal como se usan en el presente documento, las expresiones “medio de programa informático”, “medio legible por ordenador” y “medio de almacenamiento legible por ordenador” se usan para referirse en general a medios tales como el disco duro que está asociado con la unidad de disco duro 2514, el disco magnético extraíble 2518, el disco óptico extraíble 2522, así como otros medios tales como tarjetas de memoria flash, discos de vídeo digital, memorias de acceso aleatorio (RAM, *random access memory*), memorias de solo lectura (ROM, *read only memory*), y similares. Tales medios de almacenamiento legibles por ordenador se distinguen de y no se superponen con medios de comunicación. Los medios de comunicación por lo general materializan instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos en una señal de datos modulada tal como una onda portadora. La expresión “señal de datos modulada” quiere decir una señal que tiene una o más de sus características establecidas o cambiadas de una forma tal como para codificar información en la señal. A modo de ejemplo, y no de limitación, los medios de comunicación incluyen medios inalámbricos tales como acústicos, de RF, de infrarrojos y otros medios inalámbricos. También se dirigen realizaciones a tales medios de comunicación.

Tal como se ha hecho notar en lo que antecede, se pueden almacenar módulos y programas informáticos (incluyendo los programas de aplicación 2532 y otros módulos de programa 2534) en el disco duro, el disco magnético, el disco óptico, la ROM o la RAM. Tales programas informáticos también se pueden recibir por medio de la interfaz de red 2550 o la interfaz de puerto serie 2542. Tales programas informáticos, cuando son ejecutados o cargados por una aplicación, posibilitan que el ordenador 2500 implemente rasgos distintivos de realizaciones de la presente invención que se analizan en el presente documento. Por consiguiente, tales programas informáticos representan controladores del ordenador 2500.

La invención también se dirige a productos de programa informático que comprenden un soporte lógico almacenado en cualquier medio usable por ordenador. Tal soporte lógico, cuando se ejecuta en uno o más dispositivos de procesamiento de datos, da lugar a que un dispositivo o dispositivos de procesamiento de datos funcionen tal como se describe en el presente documento. Realizaciones de la presente invención emplean cualquier medio usable por ordenador o legible por ordenador, que sea conocido en la actualidad o en el futuro. Los ejemplos de medios legibles por ordenador incluyen, pero no se limitan a dispositivos de almacenamiento tales como RAM, almacenamiento flash (por ejemplo, memoria flash), unidades de estado sólido (SSD, *solid-state drive*), unidades de disco duro, discos flexibles, CD ROM, DVD ROM, discos Zip, cintas, dispositivos de almacenamiento magnético, dispositivos de almacenamiento óptico, MEM, dispositivos de almacenamiento basados en nanotecnología, y similares.

VI. Conclusión

A pesar de que se han descrito en lo que antecede varias realizaciones de la presente invención, se debería entender que estas se han presentado solo a modo de ejemplo, y no de limitación.

REIVINDICACIONES

1. Un procedimiento de copia de seguridad de flujo de datos, que comprende:

identificar para copia de seguridad una pluralidad de flujos de datos optimizados almacenados en un almacén de fragmentos (118; 300; 1334), incluyendo el almacén de fragmentos cada flujo de datos optimizado como una pluralidad de fragmentos y metadatos de flujo optimizado correspondientes, incluyendo la pluralidad de fragmentos al menos un fragmento de datos (208; 322), haciendo referencia los metadatos de flujo optimizado correspondientes al al menos un fragmento de datos, e incluyendo el almacén de fragmentos todos los fragmentos de datos incluidos de una forma desduplicada; y almacenar al menos una porción del almacén de fragmentos en un almacenamiento de copia de seguridad (1304) para realizar una copia de seguridad de la pluralidad de flujos de datos optimizados identificados para copia de seguridad, en el que dicho almacenamiento comprende:

seleccionar (1802) una técnica de copia de seguridad de entre una pluralidad de técnicas de copia de seguridad basándose en heurística, en el que una compensación recíproca en la técnica de copia de seguridad se basa en un diferencial entre un tamaño optimizado y un tamaño lógico de archivos de la pluralidad de flujos de datos optimizados identificados para copia de seguridad; y realizar (1804) la técnica de copia de seguridad seleccionada para realizar una copia de seguridad de la pluralidad de flujos de datos optimizados en el almacenamiento de copia de seguridad.

2. El procedimiento de la reivindicación 1, en el que dicho almacenamiento comprende:

almacenar el almacén de fragmentos (118; 300; 1334) en su totalidad en el almacenamiento de copia de seguridad (1304); y almacenar una pluralidad de códigos auxiliares de metadatos de flujo en el almacenamiento de copia de seguridad para la pluralidad de flujos de datos optimizados, en el que cada código auxiliar de metadatos de flujo vincula un flujo de datos con unos datos correspondientes en el almacén de fragmentos.

3. El procedimiento de la reivindicación 1, en el que dicho almacenamiento comprende:

rehidratar cada flujo de datos optimizado en un flujo de datos no optimizado correspondiente que incluye cualesquiera fragmentos de datos (208; 322) a los que hacen referencia los metadatos de flujo optimizado correspondientes; y almacenar cada flujo de datos no optimizado en el almacenamiento de copia de seguridad (1304), estando el almacén de fragmentos (118; 300; 1334) excluido de ser almacenado en el almacenamiento de copia de seguridad.

4. El procedimiento de la reivindicación 1, en el que dicho almacenamiento comprende:

recibir un primer flujo de datos optimizado identificado para copia de seguridad; determinar uno o más fragmentos de datos (208; 322) a los que hacen referencia los metadatos de flujo optimizado del primer flujo de datos optimizado y no almacenados aún en el almacenamiento de copia de seguridad (1304); almacenar los metadatos de flujo optimizado del primer flujo de datos optimizado en el almacenamiento de copia de seguridad; y almacenar los uno o más fragmentos de datos determinados en el almacenamiento de copia de seguridad.

5. El procedimiento de la reivindicación 1, en el que dicho almacenamiento comprende:

analizar los metadatos de flujo optimizado que se corresponden con cada flujo de datos optimizado para determinar al menos un identificador de fragmento de datos correspondiente para el al menos un fragmento de datos (208; 322) al que hacen referencia los metadatos de flujo optimizado; almacenar una estructura de flujo optimizado para cada flujo de datos optimizado en el almacenamiento de copia de seguridad (1304) con el al menos un identificador de fragmento de datos correspondiente; y almacenar uno o más contenedores de fragmentos del almacén de fragmentos (118; 300; 1334) en el almacenamiento de copia de seguridad que almacenan fragmentos de datos de los flujos de datos optimizados.

6. El procedimiento de la reivindicación 1, en el que dicha selección de una técnica de copia de seguridad basándose en heurística comprende:

determinar si la pluralidad de flujos de datos optimizados se seleccionaron para copia de seguridad de acuerdo con un modo de exclusión o un modo de inclusión, siendo el modo de exclusión un primer modo de copia de seguridad en el que al menos un volumen se selecciona específicamente para copia de seguridad y uno o más flujos de datos se seleccionan específicamente para excluirse de la copia de seguridad, y siendo el modo de inclusión un segundo modo de copia de seguridad en el que al menos un flujo de datos se selecciona específicamente para copia de seguridad; seleccionar la técnica de copia de seguridad para que sea una técnica de copia de seguridad de flujo de datos

optimizado si se determinara que se ha seleccionado el modo de exclusión; y seleccionar la técnica de copia de seguridad para que sea una técnica de copia de seguridad de flujo de datos no optimizado si se determinara que se ha seleccionado el modo de inclusión.

7. El procedimiento de la reivindicación 1, en el que dicha selección de una técnica de copia de seguridad basándose en heurística comprende:

determinar una primera cantidad de espacio en el almacén de fragmentos (118; 300; 1334) consumida por los fragmentos de datos (208; 322) a los que no hace referencia la pluralidad de flujos de datos optimizados identificados para copia de seguridad;
determinar una segunda cantidad de espacio como una cantidad de espacio para almacenar la totalidad de la pluralidad de flujos de datos optimizados en una forma no optimizada menos una cantidad de espacio para almacenar la totalidad de la pluralidad de flujos de datos optimizados en una forma optimizada;
seleccionar la técnica de copia de seguridad para que sea una técnica de copia de seguridad de flujo de datos optimizado si la primera cantidad determinada de espacio es menor que la segunda cantidad determinada de espacio; y
seleccionar la técnica de copia de seguridad para que sea una técnica de copia de seguridad de flujo de datos no optimizado si la primera cantidad determinada de espacio es más grande que la segunda cantidad determinada de espacio.

8. El procedimiento de la reivindicación 1, en el que dicha selección de una técnica de copia de seguridad basándose en heurística comprende:

determinar una primera cantidad de espacio como una cantidad de espacio para almacenar la totalidad de la pluralidad de flujos de datos identificados para copia de seguridad en una forma no optimizada;
determinar una segunda cantidad de espacio como una suma de una cantidad de espacio para almacenar el almacén de fragmentos (118; 300; 1334) y una cantidad de espacio para almacenar la totalidad de la pluralidad de flujos de datos identificados para copia de seguridad en una forma optimizada;
seleccionar la técnica de copia de seguridad para que sea una técnica de copia de seguridad de flujo de datos optimizado si la primera cantidad determinada de espacio es más grande que la segunda cantidad determinada de espacio; y
seleccionar la técnica de copia de seguridad para que sea una técnica de copia de seguridad de flujo de datos no optimizado si la primera cantidad determinada de espacio es menor que la segunda cantidad determinada de espacio.

9. Un producto de programa informático que comprende un medio legible por ordenador que tiene una lógica de programa informático registrada en el mismo, que comprende:

unos medios de lógica de programa informático, los cuales, cuando se ejecutan en un ordenador, realizan un procedimiento de acuerdo con cualquiera de las reivindicaciones 1 - 8.

10. Un sistema, que comprende:

un módulo de copia de seguridad de datos (1302) que está adaptado para recibir una identificación de una pluralidad de flujos de datos optimizados almacenados en un almacén de fragmentos (118; 300; 1334) para copia de seguridad, incluyendo el almacén de fragmentos cada flujo de datos optimizado como una pluralidad de fragmentos y metadatos de flujo optimizado correspondientes, incluyendo la pluralidad de fragmentos al menos un fragmento de datos (208; 322), haciendo referencia los metadatos de flujo optimizado correspondientes al al menos un fragmento de datos, incluyendo el almacén de fragmentos todos los fragmentos de datos incluidos de una forma desduplicada; y
estando configurado el módulo de copia de seguridad de datos para almacenar al menos una porción del almacén de fragmentos en un almacenamiento de copia de seguridad (1304) para realizar una copia de seguridad de la pluralidad de flujos de datos optimizados identificados para copia de seguridad, en el que el módulo de copia de seguridad de datos incluye un módulo de heurística (118; 1314) que está configurado para seleccionar (1802) una técnica de copia de seguridad de entre una pluralidad de técnicas de copia de seguridad basándose en heurística, en el que una compensación recíproca en la técnica de copia de seguridad se basa en un diferencial entre un tamaño optimizado y un tamaño lógico de archivos de la pluralidad de flujos de datos optimizados identificados para copia de seguridad, y para posibilitar que la técnica de copia de seguridad seleccionada se realice (1804) para realizar una copia de seguridad de la pluralidad de flujos de datos optimizados en el almacenamiento de copia de seguridad.

11. El sistema de la reivindicación 10, en el que el módulo de copia de seguridad de datos comprende al menos uno de un módulo de copia de seguridad de archivos optimizada (1306) configurado para almacenar el almacén de fragmentos en su totalidad en el almacenamiento de copia de seguridad (1304);
un módulo de copia de seguridad de rehidratación (1308) configurado para rehidratar cada flujo de datos optimizado en un flujo de datos no optimizado correspondiente que incluye cualesquiera fragmentos de datos (208; 322) a los que hacen referencia los metadatos de flujo optimizado correspondientes, y para almacenar cada flujo de datos no

optimizado en el almacenamiento de copia de seguridad;

un módulo de copia de seguridad a nivel de elemento (1312) configurado para cada flujo de datos optimizado para determinar uno o más fragmentos de datos a los que hacen referencia los metadatos de flujo optimizado del flujo de datos optimizado y no almacenados aún en el almacenamiento de copia de seguridad, para almacenar los metadatos de flujo optimizado del flujo de datos optimizado en el almacenamiento de copia de seguridad, y para almacenar los uno o más fragmentos de datos determinados en el almacenamiento de copia de seguridad; o un módulo de copia de seguridad de identificadores de fragmento de datos configurado para analizar los metadatos de flujo optimizado de cada flujo de datos optimizado para determinar al menos un identificador de fragmento de datos correspondiente para el al menos un fragmento de datos al que hacen referencia los metadatos de flujo optimizado, para almacenar cada flujo de datos optimizado en el almacenamiento de copia de seguridad con el al menos un identificador de fragmento de datos correspondiente, y para almacenar uno o más contenedores de fragmentos del almacén de fragmentos en el almacenamiento de copia de seguridad que almacenan fragmentos de datos de los flujos de datos optimizados.

12. El sistema de la reivindicación 10, que comprende adicionalmente:

un reconstructor de archivos (2004) que recibe una solicitud para que un flujo de datos sea recuperado de un almacén de fragmentos en el almacenamiento de copia de seguridad (1304), incluyendo la solicitud un identificador para metadatos de flujo optimizado que se corresponden con el flujo de datos; incluyendo el reconstructor de archivos un módulo de devolución de llamada (2008) que genera una primera llamada a una aplicación de restauración basándose en el identificador de metadatos de flujo optimizado, especificando la primera llamada un nombre de archivo para un primer contenedor de fragmentos en el almacenamiento de copia de seguridad que almacena metadatos de flujo optimizado identificados por el identificador de metadatos de flujo optimizado, y especificando un desplazamiento para los metadatos de flujo optimizado en el primer contenedor de fragmentos; recibiendo el reconstructor de archivos los metadatos de flujo optimizado en respuesta a la primera llamada, y determinando al menos un identificador de fragmento de datos al que se hace referencia en los metadatos de flujo optimizado; generando el módulo de devolución de llamada al menos una llamada adicional a la aplicación de restauración que se corresponde con el al menos un identificador de fragmento de datos para obtener al menos un fragmento de datos (208; 322) a partir de al menos un contenedor de fragmentos en el almacenamiento de copia de seguridad; y recibiendo el reconstructor de archivos el al menos un fragmento de datos en respuesta a la al menos una llamada adicional, y combinando el al menos un fragmento de datos para restaurar el flujo de datos.

13. El sistema de la reivindicación 12, en el que el módulo de devolución de llamada (2008) genera una llamada adicional a la aplicación de restauración (2006) para generar una lista clasificada de contenedores de fragmentos; y generando el módulo de devolución de llamada una pluralidad de llamadas adicionales a la aplicación de restauración que se corresponde con la pluralidad de fragmentos de datos (208; 322) en un orden definido por la lista clasificada.

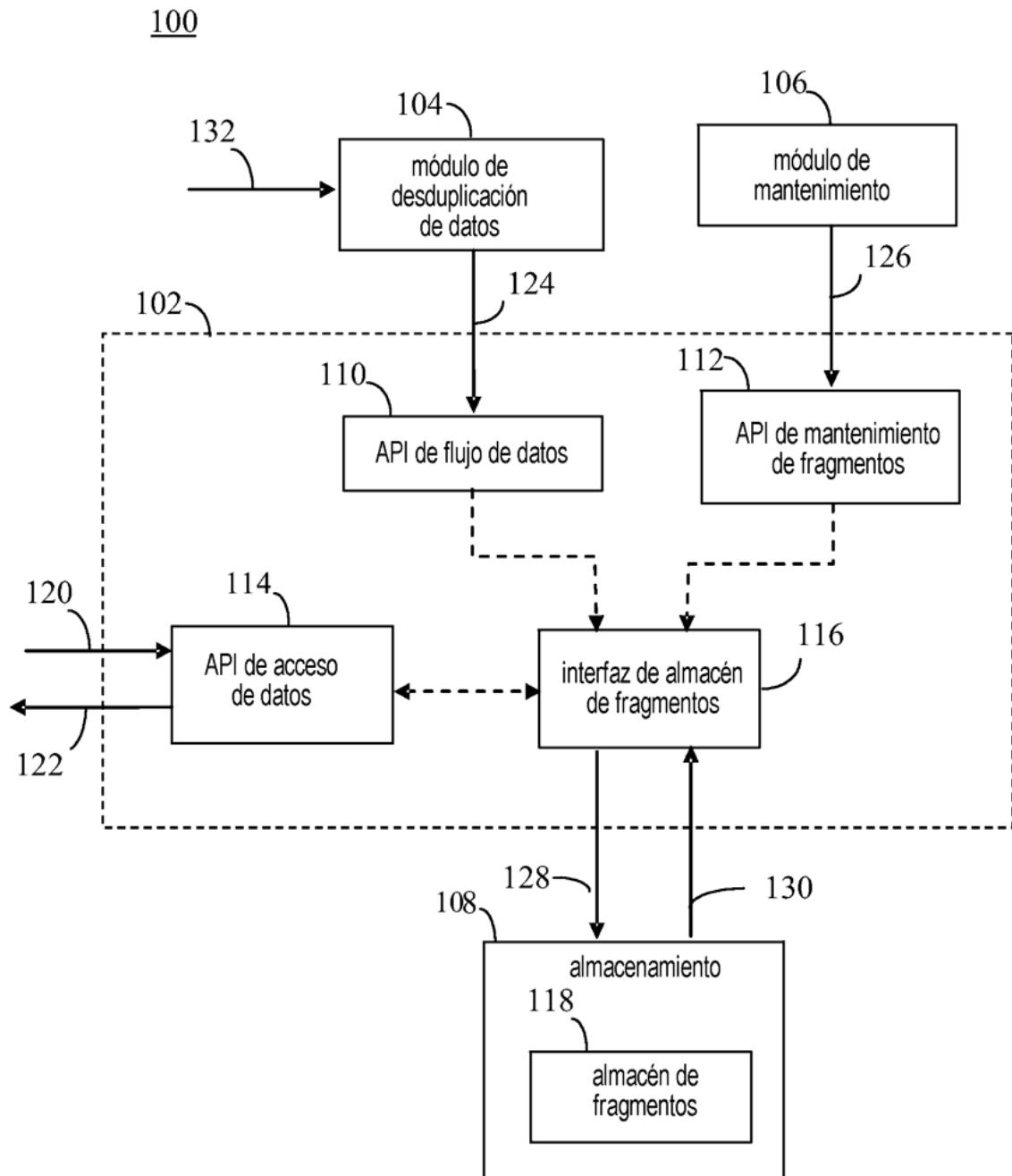


Fig.1

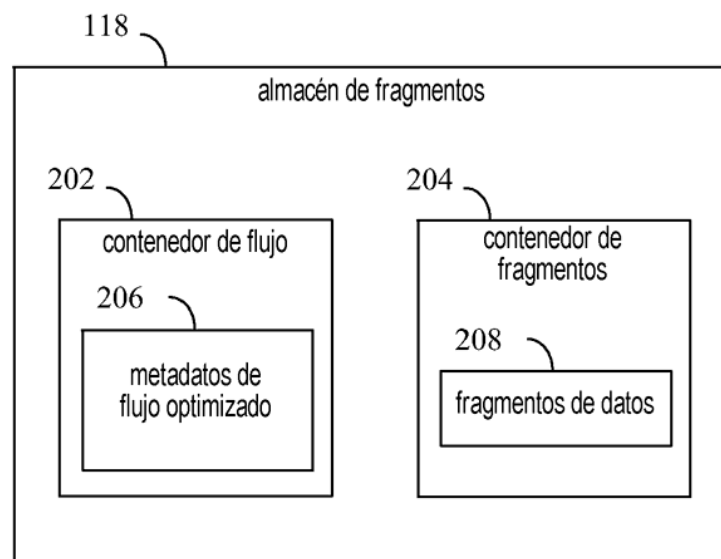


Fig.2

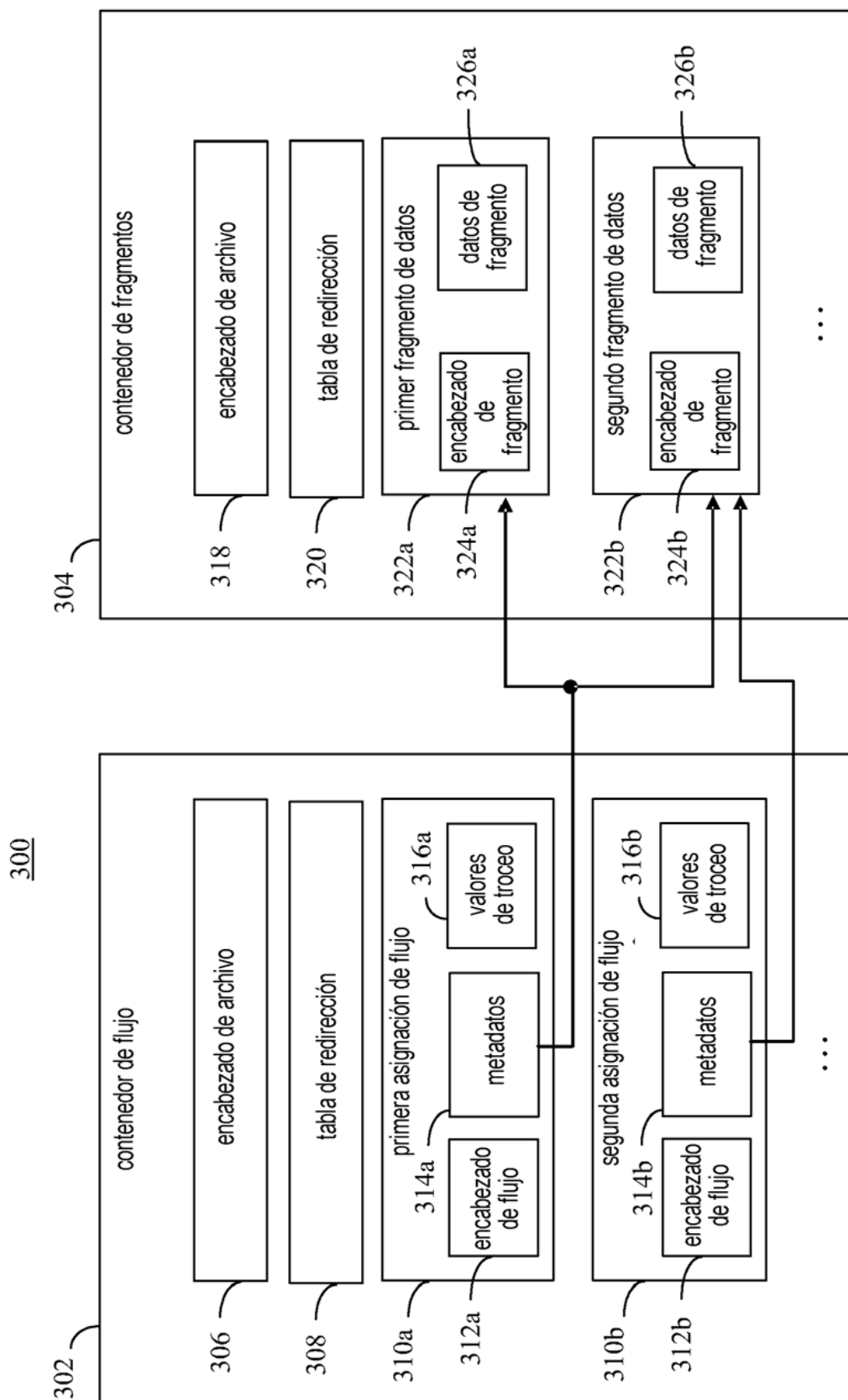


Fig.3

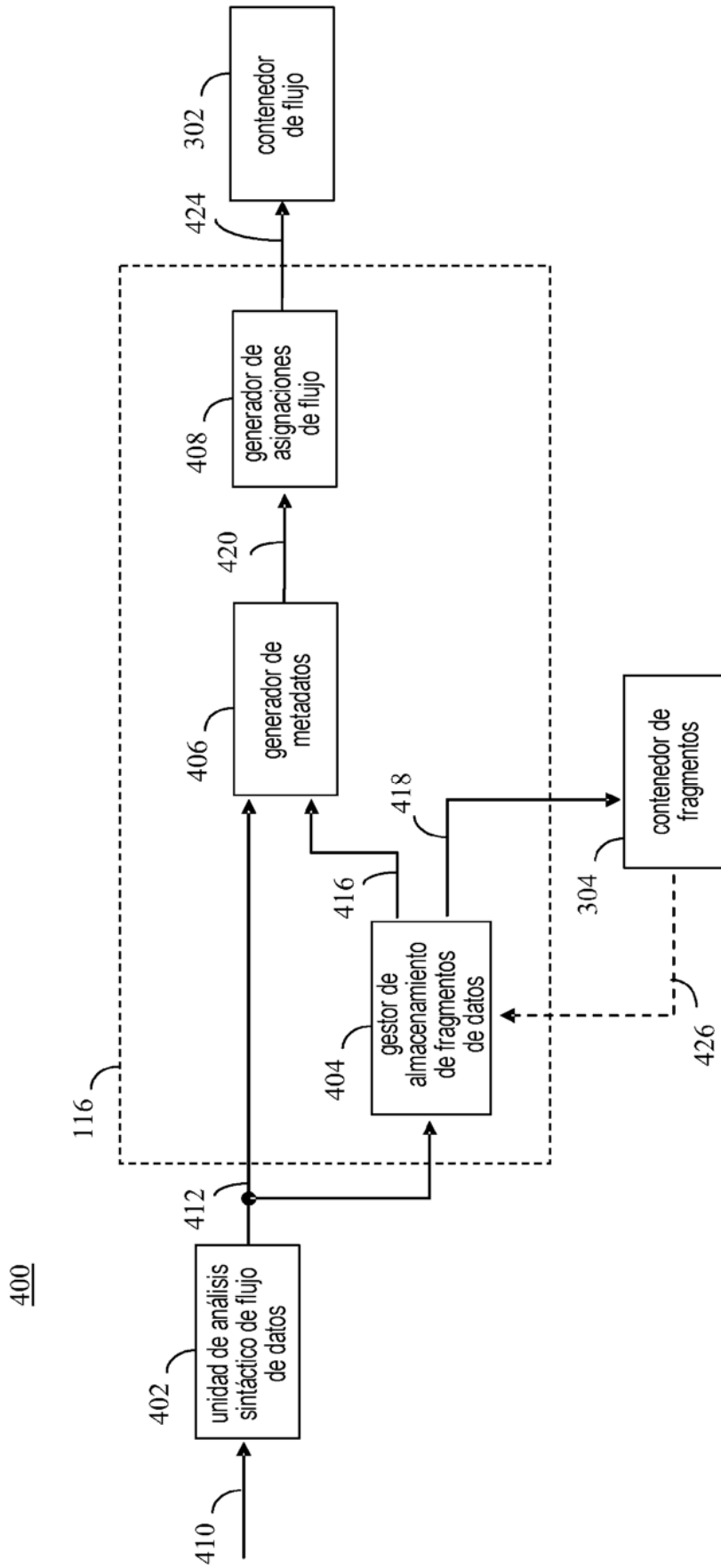
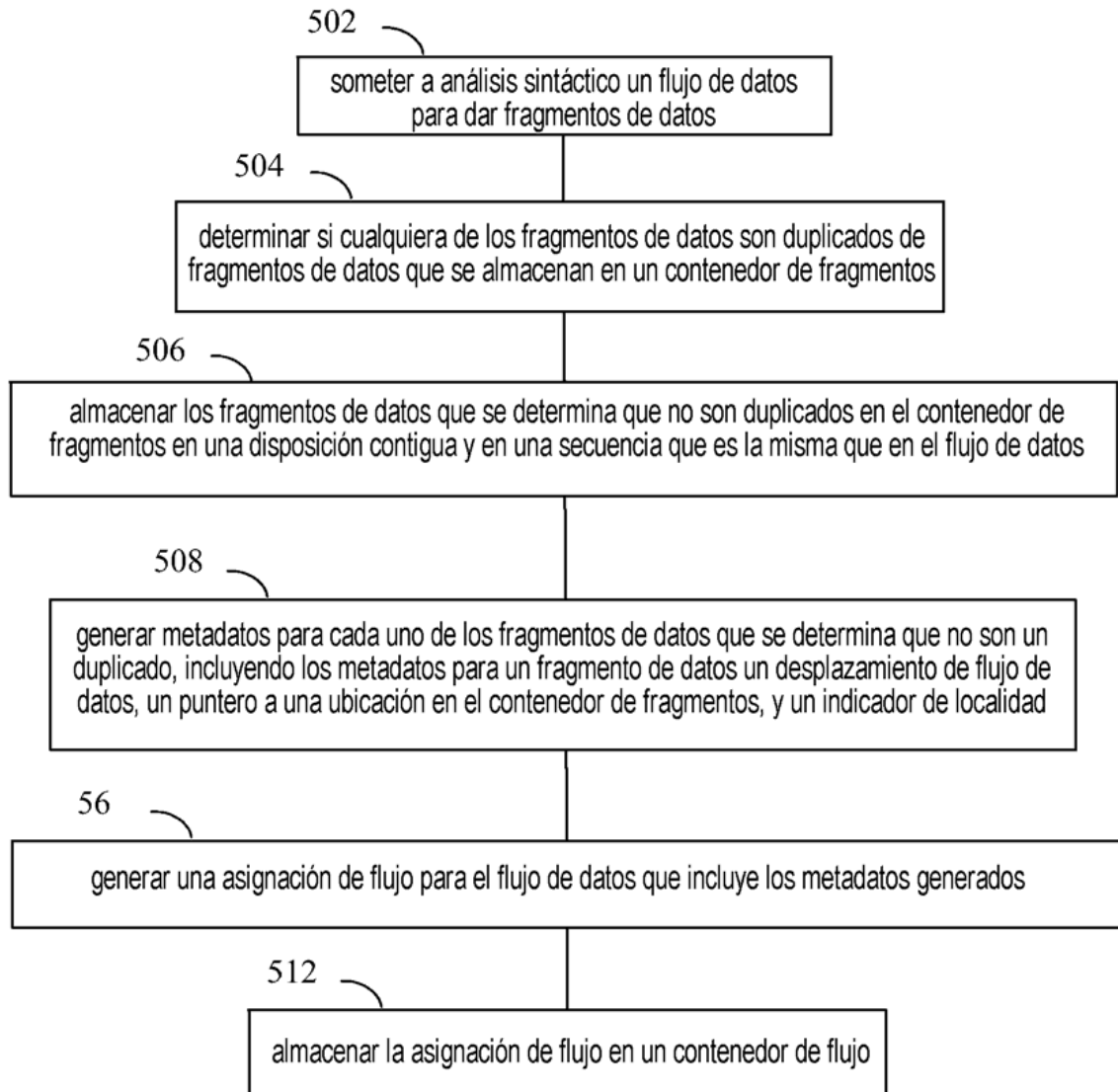


Fig.4

500**Fig.5**

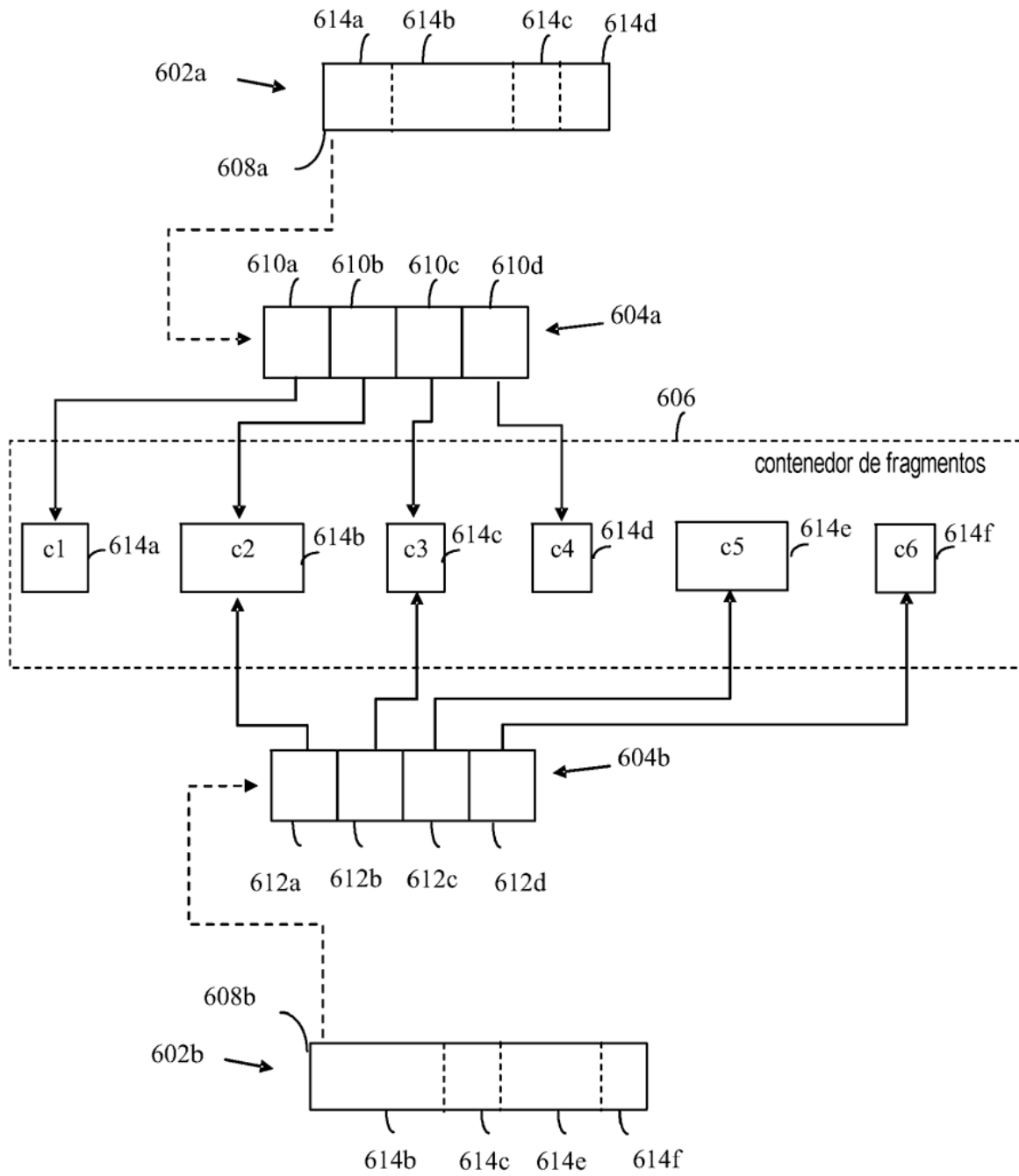


Fig.6

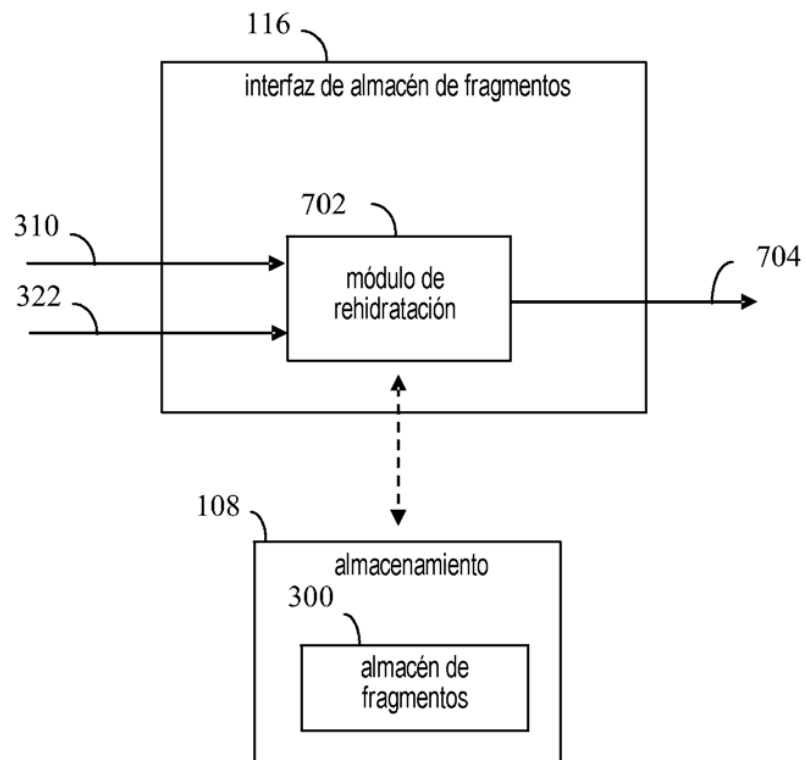


Fig.7

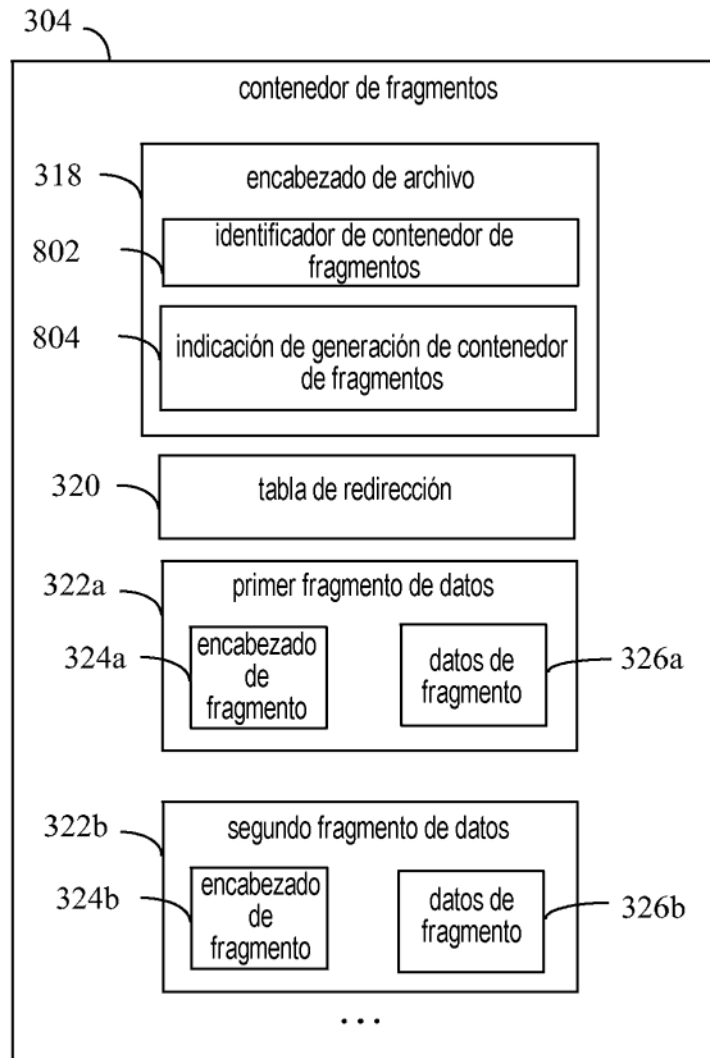


Fig.8

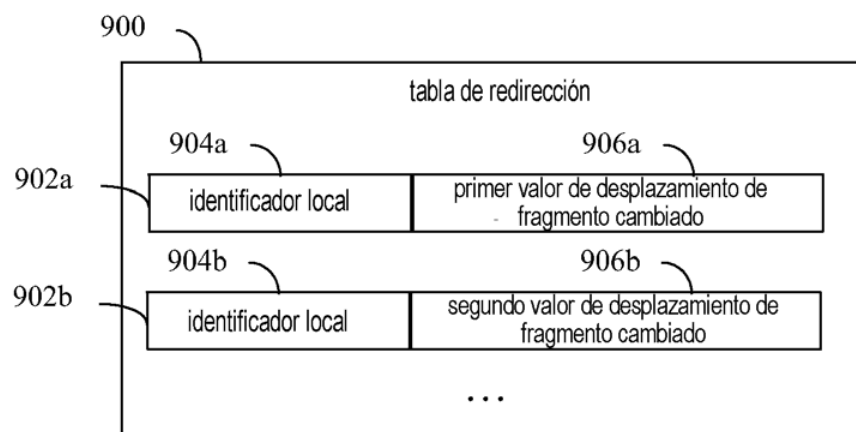


Fig.9

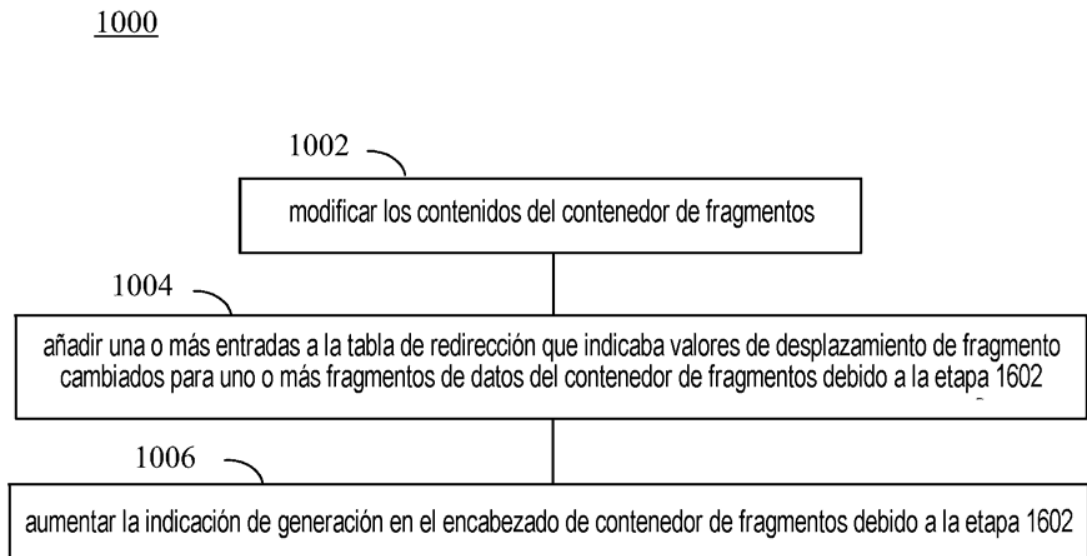


Fig.10

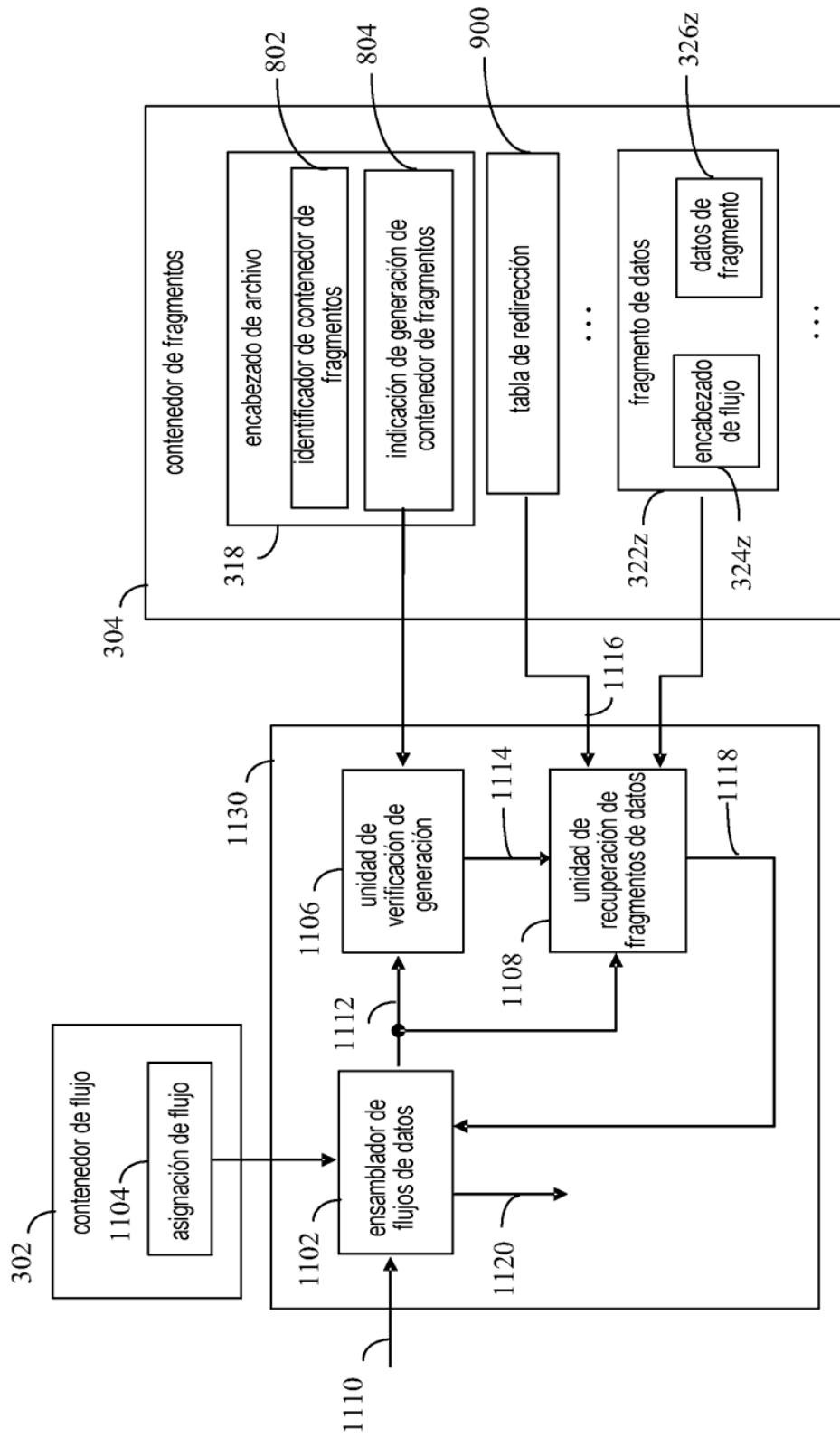


Fig.11

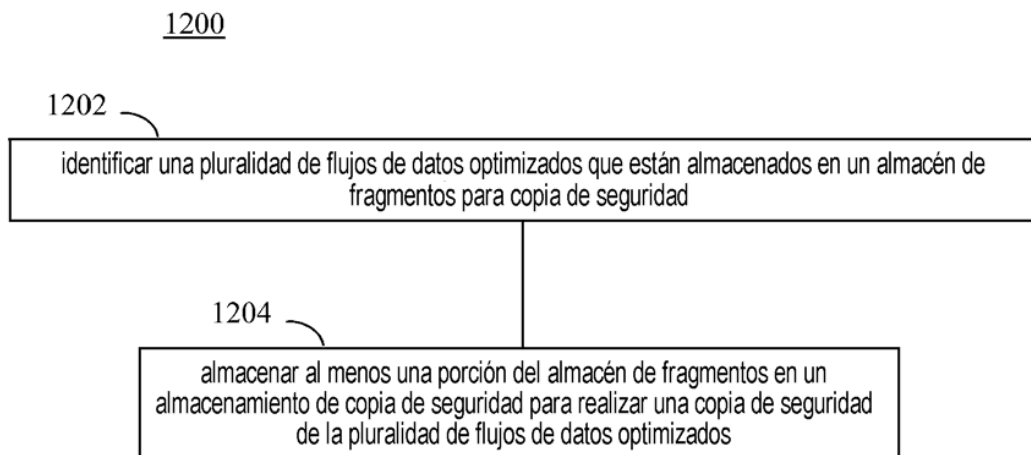


Fig.12

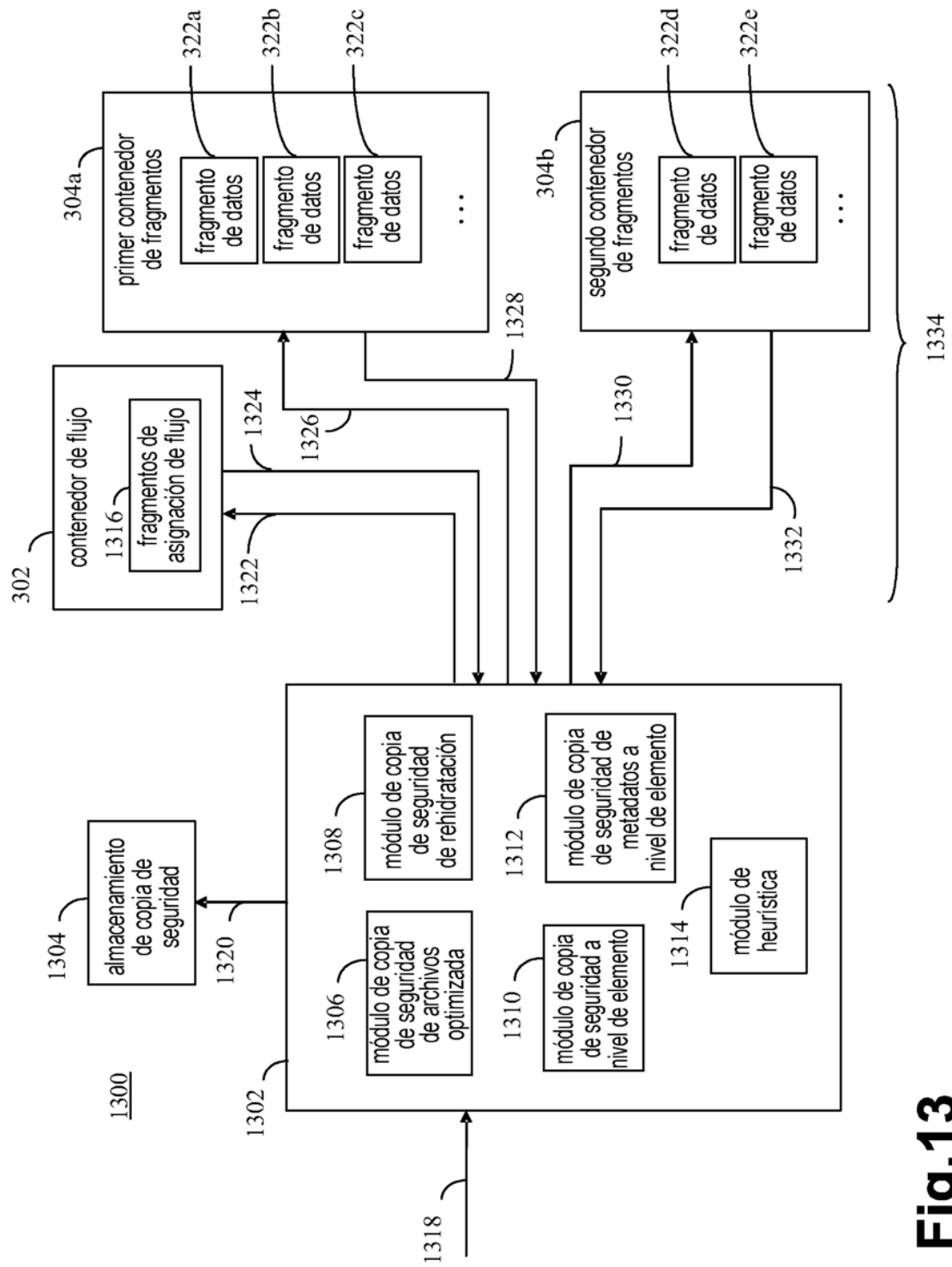


Fig.13

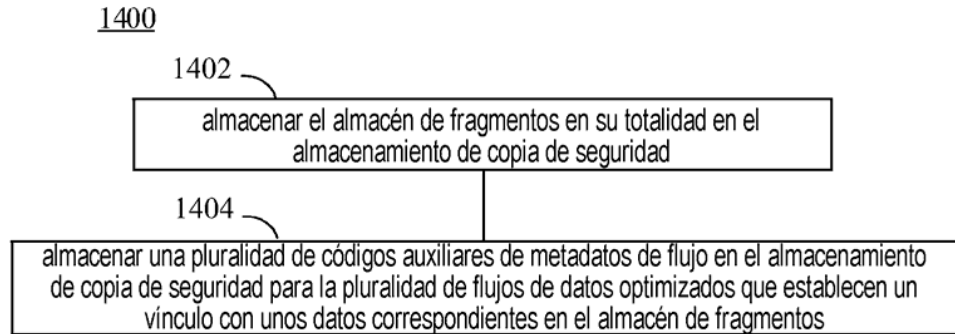


Fig.14

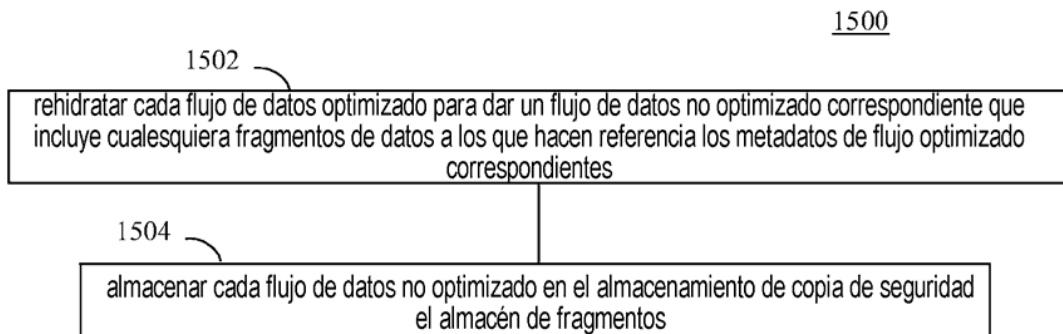


Fig.15

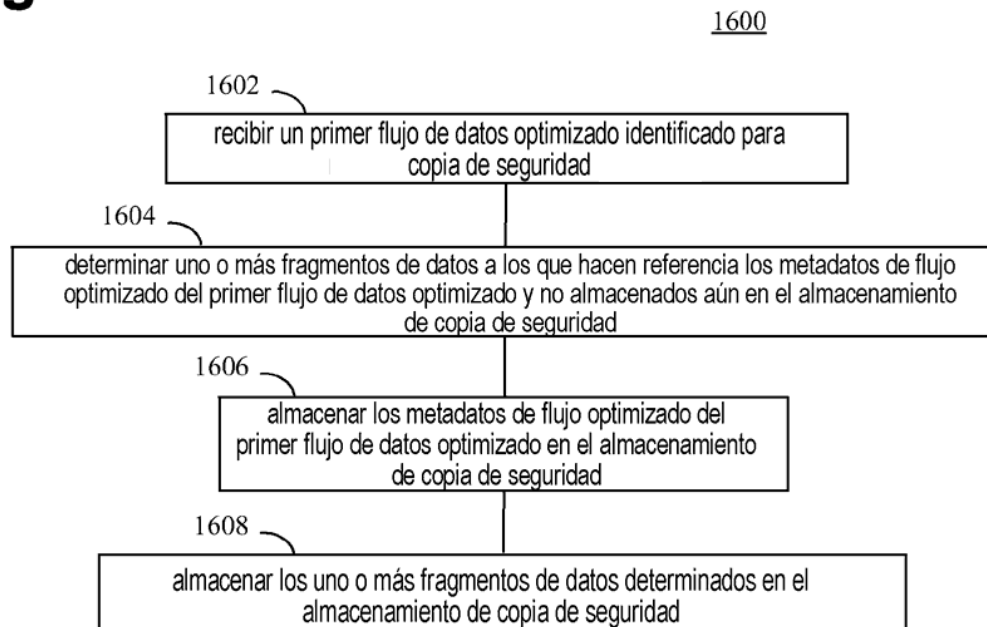


Fig.16

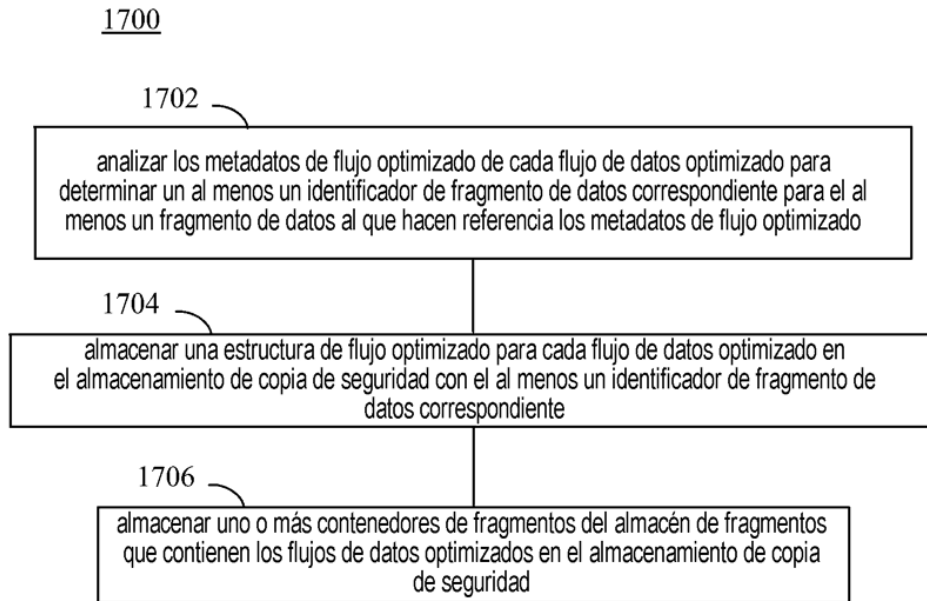


Fig.17

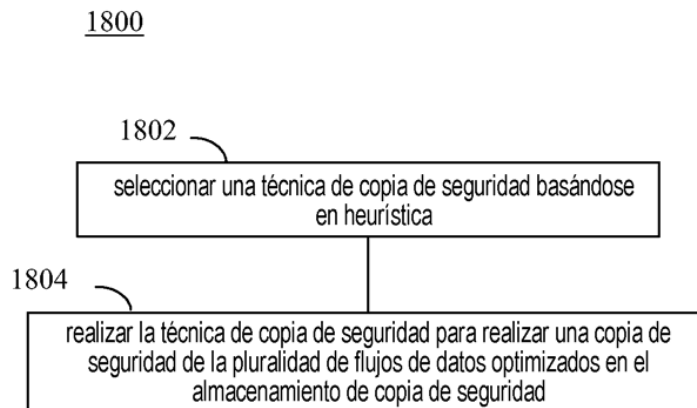
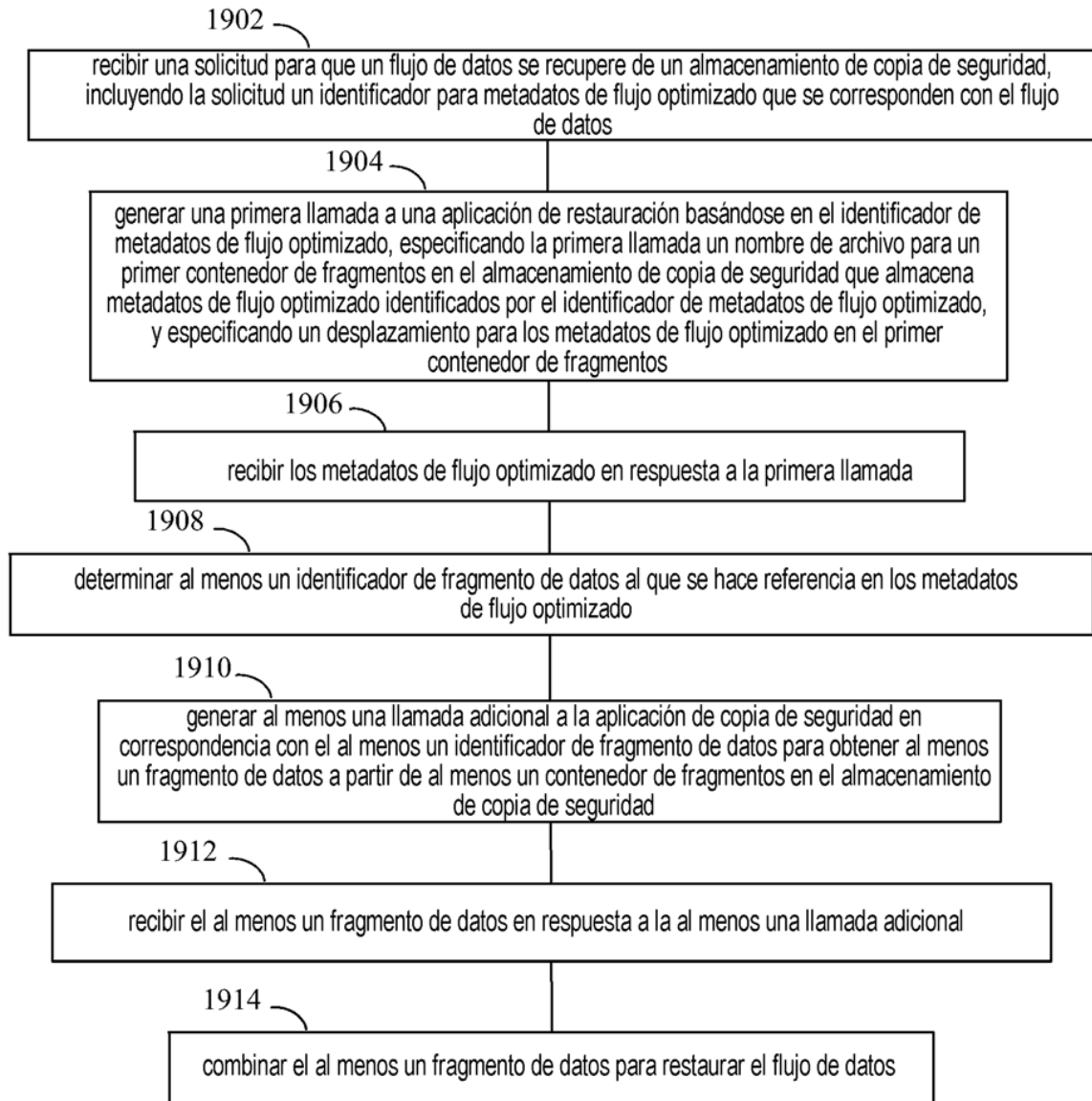


Fig.18

1900**Fig.19**

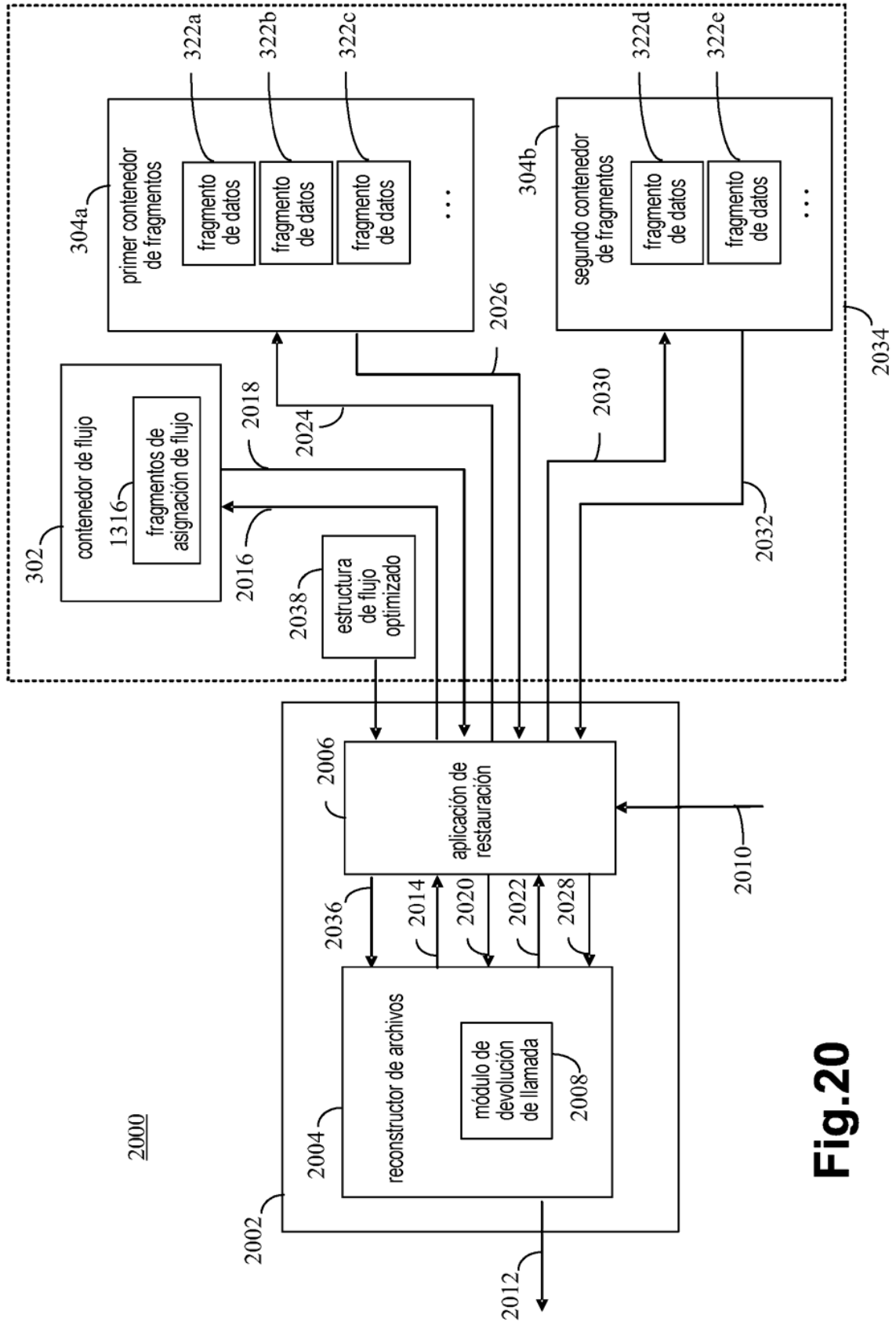


Fig. 20



Fig.21

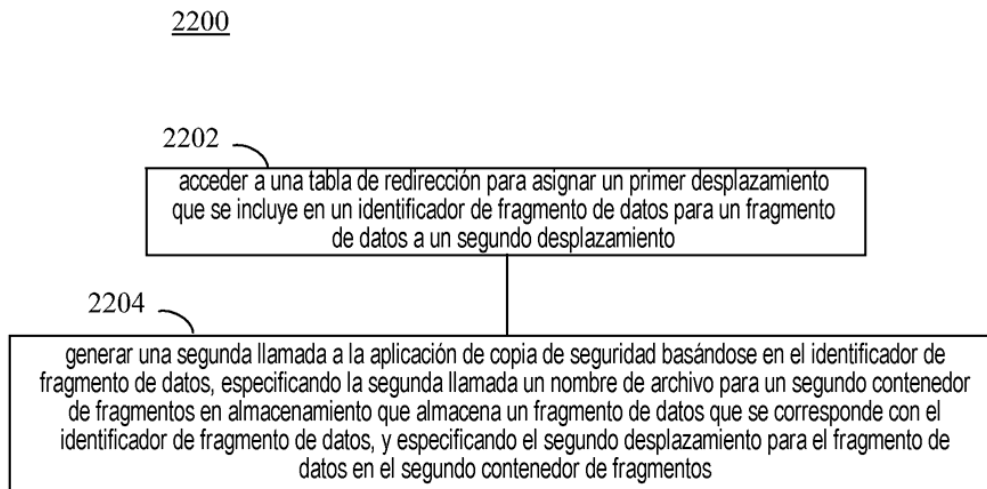


Fig.22

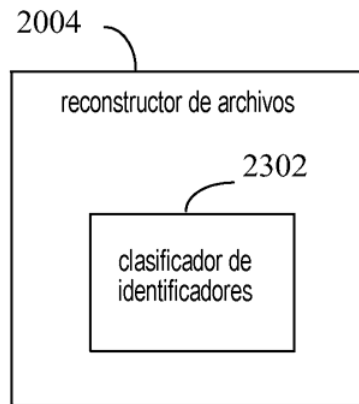


Fig.23

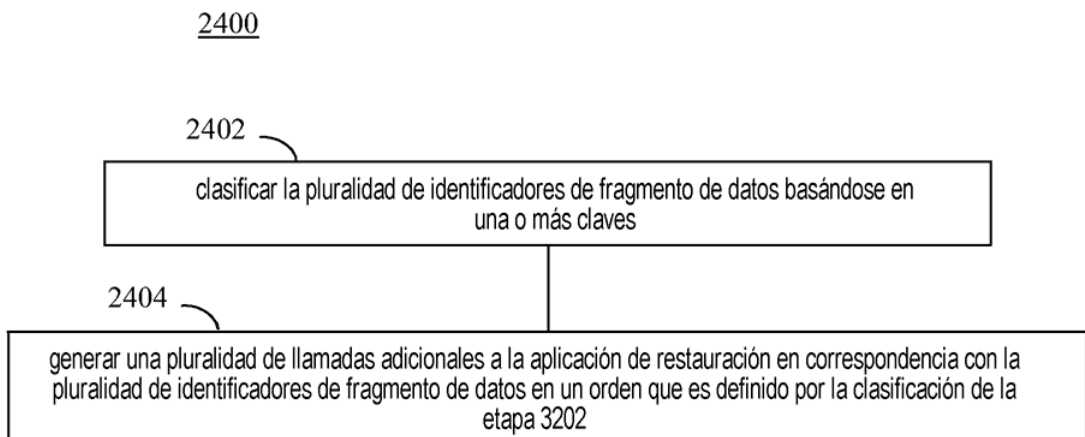


Fig.24

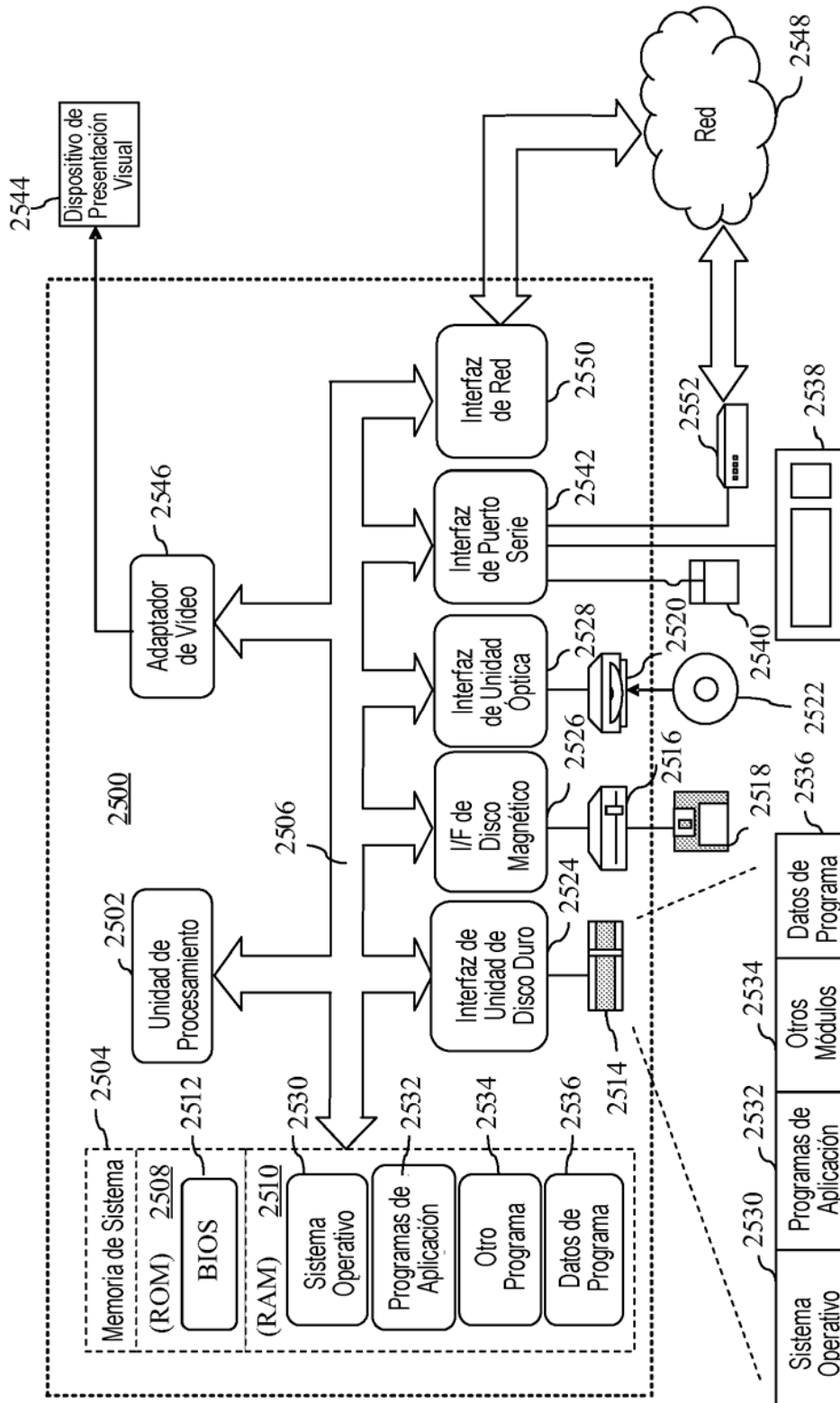


Fig.25