(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 528 245**

(51) Int. CI.:

G06F 12/00 (2006.01)
G06F 13/00 (2006.01)
G06F 11/20 (2006.01)
G06F 11/16 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **03.02.2010 E 10739071 (8)**

(97) Fecha y número de publicación de la concesión europea: **22.10.2014 EP 2394220**

(54) Título: **Almacenamiento distribuido de datos recuperables**

(30) Prioridad:

03.02.2009 US 149676 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
05.02.2015

(73) Titular/es:

BITTORRENT, INC. (100.0%)
303 Second Street, Suite S200
San Francisco, CA 94107, US

(72) Inventor/es:

BRAM, COHEN

(74) Agente/Representante:

VALLEJO LÓPEZ, Juan Pedro

DESCRIPCIÓN

Almacenamiento distribuido de datos recuperables

Remisión a solicitudes relacionadas

La presente solicitud reivindica el beneficio de la solicitud provisional U.S. 61/149.676, presentada el 3 de febrero de 2009.

Antecedentes

Sector técnico

La presente invención se refiere en general al almacenamiento distribuido de datos sobre varios nodos en una red.

Descripción de la técnica relacionada

Un problema fundamental del almacenamiento es cómo almacenar datos de manera redundante, de modo que incluso si falla un elemento de almacenamiento particular, los datos se puedan recuperar a partir de otras fuentes. Una de las estrategias consiste simplemente en almacenar múltiples copias de todo. Aunque esto funciona, exige considerablemente un mayor almacenamiento para un nivel particular de fiabilidad (o, por contraposición lógica, proporciona una fiabilidad considerablemente menor para una cantidad particular de almacenamiento).

Para lograr una fiabilidad mejor, se pueden usar códigos de borrado (*erasure codes*). Un código de borrado toma un elemento de datos original y genera a partir de él lo que se denomina “porciones”. Las porciones se diseñan de manera que siempre que haya porciones suficientes tal que su tamaño combinado sea igual al tamaño de los datos originales, los datos originales se pueden reconstruir a partir de ellas. En el esquema al que se hace referencia como codificación de borrado de *k*-*de*-*n*, se generan *n* porciones y se pueden usar *k* cualesquiera de ellas para reconstruir los datos originales. El tamaño de cada porción es 1/*k* veces el tamaño de los datos originales de manera que las porciones contienen suficiente información para la reconstrucción. La n puede variar notablemente. El almacenamiento de más porciones dará como resultado una mayor fiabilidad, aunque el número de porciones puede variar desde *k* hasta esencialmente infinito. El esquema trivial de simplemente almacenar múltiples copias de los datos originales se puede considerar como un esquema de 1-de-*n* para *n* copias, y el esquema de muy baja fiabilidad, aunque también sencillo, en el que los datos originales se trocean en fragmentos y los mismos se almacenan todos por separado se puede considerar como un esquema de *k*-de-*k* para *k* fragmentos.

La técnica de codificación de borrado (*erasure coding*) en primer lugar descompone los datos originales en *k* fragmentos, a continuación trata dichos fragmentos como vectores en un campo de Galois (GF) y genera porciones multiplicando los fragmentos por coeficientes aleatorios y sumándolos entre sí. La codificación de borrado también se puede llevar a cabo tratando los fragmentos como vectores módulo un número primo. Para simplificar, la codificación de borrado que se describe más abajo usa campos de Galois. Una porción comprende entonces el resultado de ese cálculo junto con los coeficientes aleatorios. La aleatoriedad de los coeficientes provoca que exista una probabilidad de que los datos originales sean no recuperables, esencialmente igual al inverso del número de elementos del campo de Galois que se esté usando. Por motivos de almacenamiento, un campo de Galois de tamaño 2^{32} ó 2^{64} (correspondiente al tratamiento de secciones de 4 y 8 bytes respectivamente como unidades individuales) constituye un compromiso razonable entre la carga de cálculo y la probabilidad de que por causalidad los datos sean no recuperables, siendo esto extremadamente improbable con 2^{32} y esencialmente imposible con 2^{64}.

La técnica anterior se enfrenta a limitaciones cuando se usa para un almacenamiento distribuido a través de Internet. Para el almacenamiento en Internet, el recurso escaso es el ancho de banda, y la capacidad de almacenamiento de los nodos extremos es esencialmente infinita (o por lo menos suficientemente económica como para no constituir un factor limitativo), lo cual da como resultado una situación en la que el factor limitativo sobre cualquier almacenamiento es la cantidad de ancho de banda para enviarlo. Para el almacenamiento inicial, esto da como resultado un modelo muy similar a aquel en el que el factor limitativo es la capacidad de almacenamiento; existe una sustitución unívoca de almacenamiento por ancho de banda. Sin embargo, después del almacenamiento inicial, se pueden consumir recursos significativos de ancho de banda para sustituir soportes de almacenamiento fallidos (y todos los soportes de almacenamiento acaban fallando). Normalmente, cuando el almacenamiento redundante se está llevando a cabo de manera local, por ejemplo, en una red de área local, se lleva a cabo una recuperación completa de los datos originales correspondientes a las porciones contenidas por los soportes fallidos y la misma se almacena de manera temporal, a continuación se crean porciones nuevas y las mismas se almacenan en otro elemento de soporte, y seguidamente la copia completa temporal es eliminada. Hacer lo mismo por Internet requeriría la recuperación de *k* porciones de los diversos soportes de almacenamiento restantes, la reconstrucción de los datos originales, la generación de *k* porciones nuevas, y el envío de las porciones a los soportes de almacenamiento (incluyendo los soportes de almacenamiento sustitutorios). Esto requiere 2*k* veces el tamaño de la porción que se está usando en ancho de banda a través de internet, lo cual se convierte rápidamente en inaceptable a medida que *k* crece.

El simple almacenamiento de múltiples copias de los datos es una mejor solución para evitar el uso de ancho de

banda, aunque presenta unas propiedades de fiabilidad mucho peores y representa un despilfarro por sí mismo de la escala de *n*. ya que cuantas más copias de algo es necesario guardar, más ancho de banda se usa cuando dichas copias se deterioran.

El documento US2007079082 propone un sistema de almacenamiento digital de archivos de datos en el cual archivos de datos originales a almacenar se dispersan usando alguna forma de algoritmo de dispersión de información en una serie de "fracciones" o subconjuntos de archivos, de tal manera que los datos de cada porción de archivo son menos utilizables o menos reconocibles o totalmente inútiles o totalmente irreconocibles por sí mismos excepto cuando se combinan con parte o la totalidad de las otras porciones del archivo.

El documento US2007177739 propone una técnica de duplicación de datos para proporcionar una duplicación, con codificación de borrado, de grandes conjuntos de datos sobre un conjunto de réplicas distribuidas geográficamente. La técnica utiliza un árbol de multidifusión para almacenar, reenviar, y codificar por borrado el conjunto de datos. La codificación por borrado de datos se puede llevar a cabo en varias posiciones dentro del árbol de multidifusión, incluyendo el origen, nodos intermedios, y nodos de destino. En una realización, el sistema comprende un nodo de origen para almacenar el conjunto de datos originales, una pluralidad de nodos intermedios, y una pluralidad de nodos hoja para almacenar los fragmentos de réplica únicos. Los nodos están configurados como un árbol de multidifusión para convertir los datos originales en los fragmentos de réplica únicos llevando a cabo una codificación de borrado distribuida en una pluralidad de niveles del árbol de multidifusión.

El documento US2005216813 propone un esquema de protección de archivos para contenido fijo en un archivo de datos distribuidos usando cálculos que aprovechan operadores de permutación de un código cíclico. En una realización ilustrativa, se describe una técnica de codificación de N+K para su uso con el fin de proteger datos que se están distribuyendo en una matriz redundante de nodos independientes (RAIN). Los propios datos pueden ser de cualquier tipo, y también pueden incluir metadatos del sistema.

Sumario

Según un primer aspecto de la presente invención, se proporciona un método tal como se expone en la reivindicación independiente 1 adjunta. De acuerdo con un segundo aspecto de la presente invención, se proporciona un sistema tal como se expone en la reivindicación independiente 7 adjunta.

Breve descripción de los dibujos

La FIG. 1 ilustra un entorno que incluye nodos de almacenamiento para el almacenamiento distribuido de datos, en una realización.

La FIG. 2 es un diagrama de flujo que ilustra un método para el almacenamiento distribuido de datos, en una realización.

La FIG. 3 es un diagrama de flujo que ilustra un método para dividir un archivo de datos en porciones y enviar porciones a nodos de almacenamiento, en una realización.

La FIG. 4 es un diagrama de flujo que ilustra un método de generación de porciones para un nodo de almacenamiento sustitutorio, en una realización.

La FIG. 5 es un diagrama esquemático que ilustra la generación de porciones sustitutorias para un nodo sustitutorio de entre nodos disponibles restantes, en una realización.

Las figuras representan realizaciones preferidas de la presente invención únicamente con fines ilustrativos. Los versados en la técnica reconocerán fácilmente, a partir de la siguiente descripción, que se pueden utilizar realizaciones alternativas de las estructuras y métodos ilustrados en la presente sin desviarse con respecto a los fundamentos de la invención aquí descritos.

Descripción detallada de las realizaciones preferidas

La FIG. 1 ilustra un entorno 100 que incluye nodos de almacenamiento 106 para el almacenamiento distribuido de datos, en una realización. El módulo de carga 102 contiene inicialmente un archivo de datos a almacenar en los nodos de almacenamiento. Un operador del módulo de carga puede desear almacenar el archivo en los nodos de almacenamiento por varios motivos, tales como una copia de seguridad del archivo o compartir el archivo con otros. El módulo de carga divide los archivos en porciones, que se describen de forma adicional posteriormente, y envía las porciones a los diversos nodos de almacenamiento a través de la red 104. Los nodos de almacenamiento 106 reciben las porciones y las almacenan. El módulo de descarga 110 recupera las porciones de los diversos nodos de almacenamiento y reconstruye el archivo de datos. En una realización, el módulo de descarga 110 es el mismo que el módulo de carga 102. El módulo de seguimiento 108 puede realizar un seguimiento de aspectos del sistema de almacenamiento, tales como la disponibilidad o indisponibilidad de los diversos nodos de almacenamiento y las posiciones de diversas porciones. Los nodos de almacenamiento 106 y el módulo de carga 102 y módulo de descarga pueden recibir información del módulo de seguimiento para enviar, recibir, o crear porciones.

Cualquiera de los nodos de almacenamiento 106 se puede convertir en indisponible en cualquier momento. Por ejemplo, se pueden perder los datos almacenados en un nodo de almacenamiento o se puede perder la conectividad de red del nodo. Las porciones enviadas por el módulo de carga 102 se distribuyen en los nodos de almacenamiento 106 de tal manera que uno o más de los nodos de almacenamiento 106 pueden fallar aunque el archivo de datos sigue siendo totalmente recuperable. Existe cierta redundancia entre porciones en nodos de almacenamiento diferentes. Cuando un nodo de almacenamiento 106 particular se convierte en indisponible, se puede activar un nodo de almacenamiento sustitutorio, y los nodos de almacenamiento 106 restantes generan porciones sustitutorias y envían estas porciones directamente al nodo de almacenamiento sustitutorio para almacenarlas allí. Estas porciones sustitutorias (a las que se hace referencia también como “nuevas”) se pueden construir recombinando porciones existentes en los nodos restantes. No es necesario que las porciones se envíen a una única posición (por ejemplo, el módulo de carga), se reconstruyan en el archivo original, y a continuación se vuelvan a dividir en porciones nuevas que se envían a los nodos actuales. Como consecuencia, el consumo de ancho de banda a través de la red 104 se reduce cuando se almacenan porciones en un nodo sustitutorio después de que un nodo se haya vuelto indisponible.

El módulo de carga 102, el módulo de descarga 110, el módulo de seguimiento 108, y los nodos de almacenamiento 106 pueden ser ordenadores que comprenden una CPU, memoria, un disco duro u otro dispositivo de almacenamiento, una interfaz de red, interfaces periféricas, y otros componentes bien conocidos. Los nodos de almacenamiento 106 pueden incluir grandes cantidades de espacio de almacenamiento disponible. En una realización, los nodos de almacenamiento pueden ser dispositivos de almacenamiento incorporados a la red. Aunque en la FIG. 1 se muestran solamente tres nodos de almacenamiento, puede haber muchos más. Adicionalmente, también puede haber múltiples módulos de carga 102 y módulos de descarga 110 que acceden a los nodos de almacenamiento. También puede haber múltiples módulos de seguimiento 108 (por ejemplo, puede haber módulos de seguimiento de reserva en caso de fallo del módulo de seguimiento principal).

El módulo de carga 102, el módulo de descarga 110, el módulo de seguimiento 108, y los nodos de almacenamiento 106 se comunican a través de una red 104. La red puede comprender Internet, una red de área local, redes inalámbricas, u otros tipos diversos de redes. En una realización, los nodos de almacenamiento pueden ser distantes geográficamente con respecto al módulo de carga y al módulo de descarga, y la transferencia de grandes cantidades de datos entre el módulo de carga, el módulo de descarga, y los nodos de almacenamiento a través de la red puede resultar cara. Como consecuencia, en algunas realizaciones puede resultar importante minimizar dichas transferencias de datos durante operaciones tales como la provisión de porciones a un nodo sustitutorio.

En una realización, el módulo de seguimiento 108 funciona como un sistema centralizado de mando y control explotado por un proveedor de servicios para coordinar la transferencia de datos entre módulos de carga, módulos de descarga, y nodos de almacenamiento. Puesto que a través del módulo de seguimiento no se realiza directamente ninguna transferencia de datos en bloque, sus recursos de ancho de banda y de cálculo pueden ser bajos, y se puede hacer que el mismo sea fiable usando técnicas de recuperación tras fallo (*failover*) más tradicionales. El módulo de seguimiento es responsable de mantener el seguimiento de las posiciones de porciones y nodos de almacenamiento, y de mantener el seguimiento de qué operaciones han resultado satisfactorias y cuáles han fallado. Las transferencias de datos de porciones se efectúan directamente (por ejemplo, desde el módulo de carga al nodo de almacenamiento o desde el nodo de almacenamiento al nodo de almacenamiento), de manera que el módulo de seguimiento proporciona la coordinación.

La FIG. 2 es un diagrama de flujo que ilustra un método para el almacenamiento distribuido de datos, en una realización. Inicialmente, en el módulo de carga 102 hay presente un archivo de datos a almacenar. El módulo de carga divide 202 los archivos en porciones que contienen cierto grado de redundancia. A continuación, estas porciones se envían 204 a varios nodos de almacenamiento 106 y se distribuyen de tal manera que las porciones de uno o más nodos se pueden perder aunque el archivo de datos originales todavía se puede recuperar a partir de las porciones de los nodos restantes. En algún momento, uno de los nodos de almacenamiento falla y las porciones almacenadas en ese nodo se pierden. Para mantener el mismo nivel de fiabilidad de almacenamiento, un nodo de almacenamiento 106 nuevo se designa como nodo sustitutorio. Las porciones sustitutorias a almacenar en el nodo sustitutorio se generan 206 en varios de los nodos existentes, y estas porciones se envían al nodo sustitutorio para su almacenamiento en este último. La etapa 206 puede producirse múltiples veces en la medida en la que múltiples nodos fallan y son sustituidos. En algún momento, un módulo de descarga 110 puede estar interesado en recuperar el archivo de datos. El módulo de descarga recupera 208 porciones de los nodos de almacenamiento actuales y reconstruye 210 el archivo de datos originales. Posteriormente se describen de forma más detallada realizaciones de las etapas de la FIG. 2.

La FIG. 3 es un diagrama de flujo que ilustra un método para dividir un archivo de datos en porciones y enviar porciones a nodos de almacenamiento, en una realización. Inicialmente, un módulo de carga 102 recibe 302 el número de nodos de almacenamiento 106 disponibles desde el módulo de seguimiento o un administrador del sistema u otra fuente. En la descripción posterior, el número inicial de nodos disponibles se representa con *N*. Se recibe también 304 un parámetro de fiabilidad, *x*. El parámetro de fiabilidad *x* representa el número de nodos de almacenamiento que contienen porciones de un archivo de datos que es necesario que estén disponibles para que el archivo de datos sea recuperable. *x* se selecciona en general de manera que sea menor que N de modo que el

archivo pueda seguir recuperándose en caso de que uno o más nodos se haga indisponible. En la descripción posterior, se supone que *x* es impar, aunque se pueden construir realizaciones para valores pares de x con modificaciones pequeñas. El parámetro de fiabilidad se puede recibir desde el módulo de seguimiento o un operador del módulo de carga u otra fuente. El mismo se puede ajustar de acuerdo con la política del sistema.

Se crean porciones a partir del archivo de datos usando un esquema de codificación de borrado de *k*-de-*n* descrito anteriormente en la sección de Antecedentes, donde en una realización $k = x*(x+1)/2$. En una realización, *k* es una función cuadrática de x. Cada porción tiene un tamaño de 1/*k* del archivo de datos. Se crea un total de *n* porciones, donde $n = xN$, y se envían *x* porciones a cada nodo de almacenamiento. En una realización, el archivo de datos se divide 306 en *k* fragmentos secuenciales que se procesan para generar 308 *n* porciones usando la codificación de borrado. Por ejemplo, en la etapa 306, un archivo de 10 millones de bytes se puede dividir en fragmentos secuenciales de 4.000 bytes cada uno. El procesado de la etapa 308 incluye multiplicar los fragmentos por coeficientes aleatorios y sumar los fragmentos (escalados) entre sí en varias combinaciones. Por ejemplo, cada fragmento de 4.000 bytes se puede multiplicar por un coeficiente aleatorio de 4 bytes multiplicando cada segmento de 4 bytes del fragmento por el coeficiente aleatorio (módulo 2^{32}) con el fin de generar fragmentos escalados. A continuación, estos fragmentos escalados se pueden sumar entre sí, dando como resultado una porción que es una combinación lineal de los fragmentos.

En una realización, los coeficientes aleatorios usados para generar una porción particular se incorporan al final de la porción. Con las porciones como vectores en un campo de Galois de tamaño 2^{32}, esto implica anexar un valor de 32 bits (cuatro bytes) al final de la porción para cada coeficiente. Estos coeficientes anexados posteriormente se pueden procesar de manera similar al resto de la porción simplificando la implementación de procesado de porciones. Por ejemplo, cuando la porción se multiplica por un valor particular o se suma a otra porción, el coeficiente aleatorio se puede multiplicar de manera similar por el valor y se puede sumar al coeficiente correspondiente en la otra porción. En una realización, el archivo de divide en unidades de tamaño razonable y a continuación se divide nuevamente en fragmentos más pequeños.

A continuación, las porciones se envían 310 a los nodos de almacenamiento 106 y se almacenan en los nodos. Específicamente, se envían *x* porciones a cada uno de los *N* nodos de almacenamiento usando la técnica de *k*-de-*n* antes descrita. Las porciones se distribuyen en los nodos de manera que el fallo de cualesquiera *N*-*x* nodos dejará información suficiente en las porciones restantes de los nodos disponibles restantes para reconstruir el archivo de datos. Las porciones proporcionan a los *N* nodos de almacenamiento una cierta cantidad de redundancia para aguantar una cierta cantidad de fallos de nodos, y proporcionan a cada uno de los nodos información referente a todos los fragmentos del archivo. Una optimización para la distribución inicial de porciones es que cada una de las porciones de un único nodo de almacenamiento se genere a partir de un conjunto diferente de (*x*+1)/2 fragmentos del archivo de datos. Esto proporciona al nodo de almacenamiento información referente a todos los fragmentos del archivo aunque disminuyendo la cantidad de cálculo requerida para generar las porciones. En una realización, los fragmentos iniciales del archivo de datos se dividen en cubos, y cada porción (para la distribución inicial) se crea combinando un fragmento de cada uno de los cubos.

El módulo de seguimiento 108 se puede actualizar 312 para tener conocimiento del estado actual de las porciones y los nodos de almacenamiento. El módulo de carga 102 puede proporcionar al módulo de seguimiento información que identifica cada porción generada y que identifica al nodo de almacenamiento 106 en el cual se almacenó la porción. La información de identificación de una porción puede incluir información referente a qué fragmento secuencial del archivo de datos se corresponde la fracción. El módulo de seguimiento también se puede actualizar con el estado de cada nodo de almacenamiento como disponible y que contiene porciones particulares relacionadas con el archivo de datos. El módulo de seguimiento también puede actualizar un estado global del archivo de datos como en un estado “almacenado”.

No se puede hacer que el almacenamiento inicial de datos en nodos de almacenamiento por el módulo de carga 102 sea fiable en el sentido técnico de presentar recuperación tras fallo (*failover*), puesto que el propio módulo de carga es un punto único de fallo. En una realización, cuando el módulo de carga solicita que se almacene un archivo de datos, el módulo de seguimiento 108 puede considerar que el archivo de datos está en el estado de “carga”, desde el cual puede cambiar al estado “fallido” si parece que el módulo de carga ha fallado. Si suficientes nodos de almacenamiento reciben las porciones requeridas, entonces el estado del archivo de datos se actualiza en el módulo de seguimiento al estado “almacenado”. El estado “almacenado” también puede tener dos sub-estados consultivos de “disponible” e “indisponible” en función de si el número de nodos de almacenamiento que tienen en ese momento porciones es por lo menos x. Inicialmente al producirse una carga satisfactoria, el sub-estado es “disponible” pero puede cambiar a medida que fallen los nodos según se describe posteriormente.

La FIG. 4 es un diagrama de flujo que ilustra un método de generación de porciones para un nodo de almacenamiento 106 sustitutorio, en una realización. Tal como se ha mencionado anteriormente, un nodo de almacenamiento puede fallar por varios motivos en cualquier momento. Dichos motivos pueden incluir, por ejemplo, fallo del equipo, fallo de conectividad de la red, y fallo de alimentación. Los nodos de almacenamiento también pueden ser desconectados intencionadamente, con o sin notificación de antemano. A un nodo que ha fallado o que ha sido desconectado se le hace referencia también como nodo indisponible. Cuando un nodo ha fallado, normalmente resulta útil establecer rápidamente un nodo sustitutorio de manera que se mantenga el mismo nivel de

fiabilidad en el sistema de nodos de almacenamiento. Si ha fallado un número menor que *N-x* nodos, entonces los nodos restantes todavía contienen información suficiente para recuperar el archivo de datos, y se pueden conectar nodos sustitutorios para mantener la fiabilidad del sistema.

Se recibe 402 una indicación de un nodo fallido y un nodo sustitutorio. Puede haber también múltiples nodos fallidos y nodos sustitutorios, aunque aquí, por claridad, se describe el caso de un único nodo. Esta indicación se puede recibir en el módulo de seguimiento 108. En una realización, el módulo de seguimiento comprueba periódicamente el estado de los nodos (o recibe actualizaciones de estado de los nodos) con el fin de identificar nodos fallidos. El módulo de seguimiento puede identificar un nodo sustitutorio particular de entre un grupo de nodos libres o puede recibir una indicación de un nodo sustitutorio proveniente de alguna otra fuente, tal como un administrador del sistema. La indicación de un nodo fallido o sustitutorio puede comprender una dirección de red u otro identificador del nodo.

A continuación, el módulo de seguimiento 108 emite una notificación para cualesquiera *x* de los nodos disponibles restantes sobre el nodo sustitutorio. En una realización, estos nodos pueden recibir una notificación directamente. En una realización, el módulo de seguimiento notifica al nodo sustitutorio sobre los nodos disponibles y el nodo sustitutorio inicia transferencias desde los nodos disponibles. A continuación, estos *x* nodos disponibles, a los que se hace referencia también como nodos de carga, generan 404, cada uno de ellos, una porción sustitutoria y envían 406 esta porción sustitutoria al nodo sustitutorio. Cada nodo de carga genera una porción sustitutoria multiplicando cada una de sus porciones (incluyendo los coeficientes de las porciones) por un valor de escala aleatorio y a continuación sumándolas entre sí. Por ejemplo, si la porción tiene una longitud de 400 bytes y el valor de escala aleatorio es un número de 4 bytes, la porción se divide en 100 secciones de 4 bytes y cada sección se multiplica por el valor de escala aleatorio (módulo 2^{32}). Los datos de archivo originales contenidos en las porciones siguen siendo recuperables ya que los coeficientes de las porciones se han visto afectados por los valores de escala aleatorios de la misma manera en la que lo han sido los datos de las porciones. Como consecuencia, la porción sustitutoria generada por cada nodo representa una combinación de los datos contenidos en ese momento por las porciones de ese nodo.

La FIG. 5 es un diagrama esquemático que ilustra la generación de porciones sustitutorias para un nodo sustitutorio de entre nodos disponibles restantes, en una realización. En el ejemplo mostrado en la FIG. 5, el número inicial de nodos, *N*, es 5, y el parámetro de fiabilidad *x* es 3, de manera que pueden fallar hasta dos nodos sin pérdida de datos y irrecuperable. Cada nodo contiene x porciones (en este ejemplo tres porciones). El nodo N2 ha fallado, lo cual da como resultado cuatro nodos disponibles restantes. N6 se ha identificado como nodo sustitutorio (o nuevo) para devolver el número de nodos a cinco. El módulo de seguimiento ha seleccionado los nodos disponibles N1, N4, y N5 para generar porciones sustitutorias para N6. Cada uno de los tres nodos seleccionados combina sus tres porciones para generar una única porción sustitutoria para N6. Estas porciones sustitutorias se envían directamente desde los nodos disponibles a N6 y se almacenan en N6.

El módulo de seguimiento 108 se actualiza 408 para indicar el(los) nodo(s) disponible(s) nuevo(s) y las porciones nuevas en el(los) nodo(s) nuevo(s). Los nodos de almacenamiento individuales pueden presentar un estado de “disponible” o “indisponible”. Los nodos nuevos se pueden actualizar para estar “disponibles” una vez que los mismos han descargado todas las porciones y el nodo fallido se puede actualizar a “indisponible”. En una realización, se supone que existe siempre una posibilidad de que un nodo vuelva en algún momento posterior, de manera que un nodo no se marca como “indisponible” permanentemente. Mientras un nodo de almacenamiento está intentando descargar una porción desde otros nodos o desde el módulo de carga, al mismo se le puede asignar un estado de “descarga” por parte del módulo de seguimiento.

El éxito o fallo de un intento de generar una porción combinada por parte de un nodo de almacenamiento es una valoración realizada por el nodo de almacenamiento y se informa sobre la misma al módulo de seguimiento 108. En una realización, durante la carga inicial de un archivo de datos, el módulo de seguimiento puede iniciar unas pocas transferencias de porciones a nodos de almacenamiento 106, a continuación añadir más a medida que las mismas o bien resultan exitosas o bien fallan, y detenerse si suficientes nodos tienen conjuntos completos de porciones o renunciar si fallan demasiadas de una vez. De manera similar, si se ha almacenado un archivo de datos y fallan suficientes nodos de almacenamiento de manera que la cantidad de porciones disponibles está por debajo de un umbral de seguridad, el módulo de seguimiento puede iniciar una transferencia a un nodo de almacenamiento nuevo, y a continuación volver al estado estable si esa transferencia tiene éxito o efectuar una recuperación tras fallo hacia otro nodo si la misma falla, y marcar temporalmente el fragmento particular de datos como indisponible en caso de que fallen demasiadas transferencias de una vez. En el ejemplo de la FIG. 5, supóngase que se ajusta un umbral de seguridad de cinco nodos. Si un nodo falla y el número de nodos cae por debajo de cinco, el módulo de seguimiento puede entrar en contacto con los nodos restantes para comenzar a generar porciones sustitutorias para un nodo nuevo. Si la transferencia de porciones sustitutorias al nodo nuevo falla, el módulo de seguimiento puede marcar este nodo como fallido, identificar un segundo nodo nuevo, y entrar en contacto con los nodos restantes para comenzar a generar porciones sustitutorias para el segundo nodo nuevo. Si la transferencia de porciones sustitutorias al segundo nodo nuevo falla también, el archivo de datos de las porciones se puede marcar como temporalmente indisponible para la descarga.

Las transferencias se pueden efectuar mediante la generación, por parte del nodo de almacenamiento de carga, de

una porción recombinada nueva, y el mero almacenamiento de la misma por parte del nodo de almacenamiento de descarga. Para hacer que el sistema de transferencia sea más versátil, un archivo se puede dividir en fragmentos inicialmente (en el módulo de carga) antes de quedar dividido en porciones, de manera que los x nodos que suministran porciones para un fragmento no tienen que ser los mismos que los x nodos que suministran las porciones para otro fragmento. Las transferencias se pueden finalizar desde un conjunto diferente de nodos a los nodos desde las que se iniciaron en caso de que uno de los nodos deje de funcionar a mitad de camino. Además, se pueden realizar transferencias desde cada fuente disponible de porciones simultáneamente, lo cual constituye una ventaja ya que el cuello de botella en las velocidades de transferencia de datos se encuentra generalmente en el lado del enlace ascendente, por lo que la descarga desde múltiples fuentes dará como resultado transferencias más rápidas así como un uso distribuido más uniformemente del ancho de banda. Esto a su vez da como resultado una rápida recuperación con respecto a un fallo y por tanto una mayor fiabilidad. En una realización, cuando un nodo de almacenamiento comienza una transferencia, a la misma se le proporciona el conjunto completo de recursos disponibles que se creen que están en línea por parte del módulo de seguimiento, incluyendo posiblemente el módulo de carga original, y el nodo a continuación realiza las transferencias a cualquiera que sea la velocidad que pueda de entre todas las fuentes disponibles.

El proceso antes descrito de sustitución de nodos fallidos 106 se puede repetir cualquier número de veces según sea necesario. Siguiendo este proceso, los nodos de almacenamiento pueden fallar y se pueden sustituir con el paso del tiempo, y las porciones para nodos sustitutorios pueden provenir de nodos que eran ellos mismos nodos sustitutorios en generaciones previas. Para recuperar el archivo original se pueden usar cualesquiera *x* nodos con conjuntos completos de porciones, con independencia de cuántas generaciones de fallos de nodo se hayan producido. En muchos casos prácticos, el número de nodos necesario para la recuperación es menor que x nodos. Por ejemplo, cuando se generan porciones sustitutorias para el primer nodo sustitutorio después de un almacenamiento inicial, basta con porciones de cualesquiera (*x*+1)/2 nodos.

Después de suficientes generaciones de fallos y sustituciones de nodos, el número de nodos necesario para una recuperación completa se aproximará de forma monótona a *x*, especialmente si se produce en algún momento una ocasión en la que el número de nodos disponibles desciende a *x*. Aunque en la práctica es probable que la fiabilidad sea mejor, el hecho de que sean necesarios *x* nodos como mucho, es un rendimiento mínimo riguroso del sistema. En una realización, el sistema de almacenamiento puede comenzar con menos nodos para el almacenamiento inicial del archivo de datos y puede incrementar el número de nodos con el paso de tiempo en la medida en la que se espere que la fiabilidad del sistema disminuya. Por ejemplo, después de un cierto tiempo o un cierto número de fallos y sustituciones de nodos, se puede añadir un nodo adicional al sistema. Las porciones para este nodo adicional se pueden generar de la misma manera que se generan porciones para nodos sustitutorios. No obstante, es posible que la fiabilidad del sistema pueda aumentar con el tiempo, puesto que los nodos que contienen todavía porciones para el archivo de datos habrán dispuesto de más tiempo de funcionamiento y por lo tanto será menos probable que fallen que aquellos que todavía no han sido comprobados los cuales se usan para el almacenamiento inicial. Esta consideración puede tender a equilibrar la reducción esperada de la fiabilidad, antes mencionada.

Si se proporcionan múltiples porciones sustitutorias desde un único nodo a nodos sustitutorios a través de generaciones subsiguientes de fallo de nodo, puede obtenerse como resultado una pérdida de datos. No obstante, es posible usar información adicional para verificar la integridad de porciones sustitutorias recibidas desde un nodo disponible con el fin de detectar esta pérdida de datos. En caso de que una porción sustitutoria no supere esta verificación, se puede solicitar una porción sustitutoria nueva del nodo o de otro nodo.

La integridad de una porción sustitutoria de un nodo disponible particular se puede verificar o bien recibiendo información adicional de ese nodo disponible particular o bien recibiendo información de otro nodo disponible. Esta verificación no hace que aumente significativamente el uso del ancho de banda a través de la red. Para verificar una porción recién recibida, un nodo sustitutorio (o nuevo) envía un valor semilla a un generador de números seudo-aleatorios (PRNG) para un nodo disponible. El nodo disponible genera una nueva porción recombinada y a continuación multiplica todas las entradas del mismo (aparte de los coeficientes) por el valor actual correspondiente de la salida del PRNG, y suma todos ellos entre sí. El nodo disponible envía esta suma, junto con los coeficientes, al nodo sustitutorio como información de verificación, a la que se hace referencia también como información de comprobación *hash*. A continuación, el nodo sustitutorio puede calcular cuál debería ser el valor de la suma, dadas las porciones completas de las que dispone y los coeficientes. El nodo sustitutorio verifica la integridad de la porción recién recibida si el valor calculado es igual a la suma recibida. Es posible que los datos disponibles no contengan ninguna redundancia, y, como resultado, la verificación de la integridad puede ser de valor limitado. No obstante, esta posibilidad es muy remota, especialmente poco después del almacenamiento inicial.

La eficacia del planteamiento antes descrito para generar porciones sustitutorias recombinando porciones en nodos disponibles se puede demostrar empíricamente. Una manera de determinar cuántos datos son recuperables a partir de un conjunto de porciones después de una serie de recombinaciones consiste en ejecutar una simulación que representa cada porción mediante un vector aleatorio. A continuación, se puede llevar a cabo una serie de operaciones sobre dichos vectores, y se puede calcular el rango de la matriz formada por los vectores correspondientes. Este rango indica la cantidad de datos recuperables a partir de los vectores (o porciones).

A continuación se presenta un análisis de la eficiencia del planteamiento antes descrito. Puesto que son necesarias

x entidades pares para una recuperación, el mejor rendimiento posible sobre la base de límites teóricos de información sería que dichas entidades pares tuviesen *k*, ó *x**(*x*+1)/2, porciones. Con este esquema, cada una de dichas *x* entidades pares tiene *x* porciones, para un total de *x***x* porciones. La relación entre las dos es por lo tanto (*x**(*x*+1)/2)/(*x***x*) = (*x*+1)/(2**x*) que, asintóticamente, se aproxima a 1/2 a medida que *x* se hace mayor, con lo que se puede hacer que *x* sea arbitrariamente grande con una carga adicional muy pequeña del ancho de banda. Puesto que una *x* mayor se corresponde con una mayor fiabilidad, es deseable que *x* se seleccione de manera que sea un valor grande (por ejemplo, ligeramente menor que el número total de nodos *N*). En la práctica es probable que el factor limitativo sobre *x* sea la carga de cálculo, aunque hay algunos otros efectos de segundo orden que podrían resultar prominentes. Son ejemplos de ellos los efectos sobre la carga de los protocolos alámbricos, la carga de indexación de datos centralizada, el hecho de que los coeficientes de tamaño se hagan significativos, o el aproximamiento de *x* al número total de nodos *N* en el sistema como conjunto, lo cual es un límite estricto.

Al generar porciones sustitutorias a partir de nodos existentes según se ha descrito anteriormente, se pueden mantener niveles altos de fiabilidad aunque conservando un uso bajo del ancho de banda. Es posible simplemente esperar hasta que queden solo *k* nodos de almacenamiento y a continuación llevar a cabo una recuperación (y una nueva generación de porciones para re-almacenar los datos en los nodos de almacenamiento). Este planteamiento presenta unas características de fiabilidad deficientes puesto que un fallo de un nodo de almacenamiento adicional puede dar como resultado pérdida de datos. Este planteamiento requiere también la regeneración del archivo original y porciones nuevas (por ejemplo, en el módulo de carga). Otro planteamiento consiste en realizar una recuperación y regeneración completa de porciones (por ejemplo, en el módulo de carga) cada vez que se pierde un nodo de almacenamiento. Esto presenta mejores características de fiabilidad, aunque puede que requiera grandes cantidades de ancho de banda.

Volviendo a la FIG. 2, un módulo de descarga 110 recupera 208, de los nodos de almacenamiento, porciones. Un módulo de descarga interesado en un archivo de datos puede recuperar información del módulo de seguimiento 108 en relación con qué nodos de almacenamiento tienen en ese momento porciones del archivo. A continuación, el módulo de descarga puede recuperar directamente las porciones de los nodos de almacenamiento 106 especificados. El módulo de descarga puede recuperar porciones de *x* o más nodos disponibles. A continuación, el módulo de descarga reconstruye 210 el archivo de datos original a partir de las porciones recuperadas. La reconstrucción se lleva a cabo usando los coeficientes incluidos con cada porción junto con información de secuencia de porciones del módulo de seguimiento.

Las descargas en general no se pueden hacer fiables en el sentido de recuperación tras fallo ya que el propio módulo de descarga es un punto único de fallo. La información de descarga se puede considerar como consultiva, y el módulo de seguimiento puede proporcionar simplemente información sobre todos los nodos disponibles que tienen porciones completas para un posible módulo de descarga, incluyendo posiblemente el módulo de carga original, y los módulos de descarga determinan para sí mismos si tienen éxito o fallan (aunque un fallo por cualquier motivo diferente a la desconexión del módulo de carga debería ser poco frecuente).

A diferencia de cuando un nodo nuevo se está convirtiendo en disponible, es aceptable que un nodo efectúe una recuperación completa para descargar múltiples porciones de un nodo único. Supóngase que un nodo de descarga (por ejemplo, el módulo de descarga 110) ya ha recibido una porción de un nodo de almacenamiento 106 de carga. Para determinar si una porción recombinada adicional del nodo de carga podría resultar útil, el nodo de carga puede enviar al nodo de descarga los coeficientes de la porción adicional. El nodo de descarga puede calcular si esa porción es linealmente independiente de las correspondientes que ya tiene usando los coeficientes, y en caso afirmativo, el nodo de descarga puede solicitar la porción del nodo de carga. En una realización, no es necesario que el nodo de descarga solicite la misma porción, sino que puede solicitar en su lugar otra porción, ya que, en general, una porción adicional será útil si y solo si todas las fracciones, excepto una minúscula, de otras porciones son útiles. El nodo de descarga también puede solicitar información de comprobación *hash* usando la misma técnica que la usada por nodos nuevos para verificar la integridad de los datos según se ha descrito anteriormente. En general, la información de comprobación *hash* de un nodo es útil si y solo si porciones adicionales de ese nodo pudieran ser útiles.

En una realización, las técnicas de almacenamiento distribuido antes descritas se aplican sobre dispositivos de almacenamiento ampliamente distribuido, no fiables (posiblemente conectados a ordenadores). En la actualidad existen muchos dispositivos de almacenamiento con cantidades muy grandes de capacidad no utilizada sobre conexiones de Internet no reguladas que se pueden usar para un almacenamiento distribuido. Estos dispositivos de almacenamiento (es decir, nodos de almacenamiento 106) pueden ser propiedad de varios negocios e individuos y pueden ser explotados por los mismos. Los dispositivos de almacenamiento pueden presentar tasas de fallo elevadas, aunque los fallos no presentan en gran medida ninguna correlación. Puesto que es improbable que muchos nodos fallen al mismo tiempo, es probable que el planteamiento antes descrito funcione bien. En una realización, el módulo de seguimiento 108 puede permitir que proveedores de dispositivos de almacenamiento hagan que su almacenamiento resulte disponible para usuarios que necesitan almacenar datos. El módulo de seguimiento puede incluir un servidor web para que proveedores de dispositivos de almacenamiento registren sus dispositivos de almacenamiento en el módulo de seguimiento, de manera que los dispositivos pueden comenzar a usarse como nodos de almacenamiento iniciales o nodos sustitutorios para archivos de datos de usuarios del

almacenamiento. El módulo de seguimiento también puede realizar un seguimiento del uso de los dispositivos de almacenamiento y facilitar el pago de usuarios del almacenamiento a proveedores de dispositivos de almacenamiento.

En la memoria descriptiva, la referencia a “una realización” significa que un rasgo, estructura, o característica particular descrito en relación con las realizaciones está incluido en por lo menos una realización de la invención. Las apariciones de la expresión “en una realización” o “una realización preferida” en varios lugares de la memoria descriptiva no se refieren todas ellas necesariamente a la misma realización.

Algunas partes de lo anterior se presentan en términos de métodos y representaciones simbólicas de operaciones sobre bits de datos en una memoria de ordenador. Estas descripciones y representaciones son los medios usados por aquellos versados en la técnica para comunicar, de manera más efectiva, la esencia de su trabajo a otros versado en la técnica. En este caso, y en general, se considera que un método es una secuencia autónoma de etapas (instrucciones) que conducen a un resultado deseado. Etapas son aquellas que requieren manipulaciones físicas de magnitudes físicas. Habitualmente, aunque no de forma necesaria, estas magnitudes adoptan la forma de señales eléctricas, magnéticas u ópticas con capacidad de ser almacenadas, transferidas, combinadas, comparadas y manipuladas de otro modo. En ocasiones resulta cómodo, principalmente por motivos de uso común, referirse a estas señales como bits, valores, elementos, símbolos, caracteres, términos, números, o similares. Además, también resulta cómodo en ocasiones, referirse a ciertas disposiciones de etapas que requieren manipulaciones físicas de magnitudes físicas como módulos o dispositivos de código, sin pérdida de generalidad.

No obstante, debe tenerse en mente que todos estos términos y similares están destinados a asociarse a las magnitudes físicas apropiadas y son meramente etiquetas adecuadas que se aplican a estas magnitudes. A no ser que se mencione específicamente lo contrario tal como se pone de manifiesto a partir de la siguiente argumentación, se aprecia que en la totalidad de la descripción, las argumentaciones que utilizan términos tales como “procesar” o “calcular” o “determinar” o “visualizar” o “determinar” o similares, se refieren a la acción y procesos de un sistema de ordenador, o dispositivo informático electrónico similar, que manipula y transforma datos representados en forma de magnitudes (electrónicas) físicas dentro de las memorias o registros del sistema informático u otros dispositivos de almacenamiento, transmisión o visualización de información del tipo mencionado.

Ciertos aspectos de la presente invención incluyen etapas de proceso e instrucciones descritas en la presente en forma de un método. Debe observarse que las etapas de proceso e instrucciones de la presente invención se pueden materializar en software, microprogramas o hardware, y cuando se materialicen en software, se pueden descargar para residir en y ser explotadas desde diferentes plataformas usadas por varios sistemas operativos.

La presente invención se refiere también a un aparato para llevar a cabo las operaciones de la presente. Este aparato se puede construir especialmente para los fines requeridos, o puede comprender un ordenador de propósito general activado o reconfigurado selectivamente por un programa de ordenador almacenado en el ordenador. Dicho programa de ordenador se puede almacenar en un soporte de almacenamiento legible por ordenador, tal como, aunque sin carácter limitativo, cualquier tipo de disco incluyendo discos flexibles, discos ópticos, CD-ROMs, discos magnético-ópticos, memorias de solo lectura (ROMs), memorias de acceso aleatorio (RAMs), EPROMs, EEPROMs, tarjetas magnéticas u ópticas circuitos integrados de aplicación específica (ASICs), o cualquier tipo de soporte adecuado para almacenar instrucciones electrónicas, y acoplado, cada uno de ellos, a un bus de un sistema de ordenador. Además, los ordenadores a los que se hace referencia en la memoria descriptiva pueden incluir un único procesador o pueden ser arquitecturas que utilizan diseños de procesador múltiple para incrementar la capacidad de cálculo.

Los métodos y representaciones visuales presentados en este documento no están relacionados inherentemente con ningún ordenador particular u otro aparato. Varios sistemas de propósito general también se pueden usar con programas de acuerdo con las enseñanzas de la presente, o puede que resulte conveniente construir aparatos más especializados para llevar a cabo las etapas de método requeridas. La estructura requerida para varios de estos sistemas se pondrá de manifiesto a partir de la siguiente descripción. Adicionalmente, la presente invención no se describe en referencia a ningún lenguaje de programación particular. Se apreciará que se puede usar varios lenguajes de programación para implementar las enseñanzas de la presente invención que se describen en la presente, y cualesquiera referencias posteriores a lenguajes específicos se proporcionan para dar a conocer la habilitación y el modo óptimo de la presente invención.

Aunque la invención se ha mostrado y descrito particularmente en referencia a una realización preferida y varias realizaciones alternativas, aquellos versados en la técnica pertinente entenderán que en la misma se pueden realizar diversos cambios en cuanto a forma y detalles sin desviarse con respecto al alcance de la invención.

Finalmente, debe señalarse que el lenguaje usado en la memoria descriptiva se ha seleccionado principalmente con fines de legibilidad e instructivos, y puede no haber sido seleccionado para delimitar o circunscribir la materia objeto de la invención. Por consiguiente, la exposición de la presente invención está destinada a ser ilustrativa, aunque no limitativa, del alcance de la invención.

REIVINDICACIONES

1. Método para sustituir un nodo fallido que almacena datos distribuidos, que comprende:

recibir (402) una indicación de un nodo de almacenamiento nuevo, recibiéndose la indicación en cada uno de una pluralidad de nodos de almacenamiento (106), en donde dicho cada uno de la pluralidad de nodos de almacenamiento (106) contiene una pluralidad de porciones generadas a partir de un archivo de datos, generándose cada una de la pluralidad de porciones mediante la multiplicación de fragmentos del archivo de datos por coeficientes aleatorios y mediante la suma de los fragmentos multiplicados:

generar (404) una porción sustitutoria en cada uno de la pluralidad de nodos de almacenamiento (106) como respuesta a la indicación, en donde la porción sustitutoria generada en uno de la pluralidad de nodos de almacenamiento incluye una combinación de la pluralidad de porciones contenidas en el nodo de almacenamiento; y

enviar (406) las porciones sustitutorias generadas desde la pluralidad de nodos de almacenamiento (106) al nodo de almacenamiento nuevo indicado.

2. El método según la reivindicación 1, en el que la porción sustitutoria generada comprende una combinación lineal de la pluralidad de porciones contenidas en el nodo de almacenamiento que usan coeficientes aleatorios, y/o en donde un número total de las porciones sustitutorias generadas por la pluralidad de nodos de almacenamiento (106) es igual a un parámetro de fiabilidad, en donde el parámetro de fiabilidad se basa en un número total de nodos de almacenamiento requeridos para que el archivo de datos siga siendo recuperable.

3. El método según una cualquiera de las reivindicaciones anteriores, que comprende además verificar la integridad de las porciones sustitutorias generadas.

4. El método según una cualquiera de las reivindicaciones anteriores, en el que:

cada una de la pluralidad de porciones generadas a partir del archivo de datos comprende una combinación lineal de fragmentos del archivo de datos y un conjunto de coeficientes usados para generar la combinación; y/o

la pluralidad de porciones generadas a partir del archivo de datos se genera usando codificación de borrado (*erasure coding*) basada en fragmentos del archivo de datos; y/o

las porciones sustitutorias generadas se usan para reconstruir el archivo de datos (210) o en donde las porciones sustitutorias generadas se usan para generar porciones sustitutorias subsiguientes para un nodo nuevo subsiguiente; y/o

la indicación se recibe desde un módulo de seguimiento (108) que monitoriza el estado de la pluralidad de nodos de almacenamiento y realiza un seguimiento de posiciones de porciones.

5. Método según la reivindicación 1, en el que el nodo de almacenamiento nuevo sustituye a un nodo de almacenamiento fallido.

6. Método según la reivindicación 1, en el que la generación de una porción sustitutoria comprende:

multiplicar cada una de la pluralidad de porciones contenidas en el nodo de almacenamiento y un conjunto de coeficientes por un valor de escala aleatorio; y

combinar las porciones multiplicadas y el conjunto multiplicado de coeficientes.

7. Sistema para sustituir un nodo fallido que almacena datos distribuidos, presentando el sistema una pluralidad de nodos de almacenamiento (106), en donde cada nodo de almacenamiento contiene una pluralidad de porciones generadas a partir de un archivo de datos, generándose cada una de la pluralidad de porciones por la multiplicación de fragmentos del archivo de datos por coeficientes aleatorios y la suma de los fragmentos multiplicados, y en donde cada nodo de almacenamiento está configurado para

recibir (402) una indicación de un nodo de almacenamiento nuevo;

generar (404) una porción sustitutoria como respuesta a la indicación, en donde la porción sustitutoria comprende una combinación de la pluralidad de porciones contenidas en el nodo de almacenamiento; y

enviar (406) la porción sustitutoria generada al nodo de almacenamiento nuevo indicado.

8. El sistema según la reivindicación 7, en el que:

la porción sustitutoria comprende una combinación lineal de la pluralidad de porciones contenidas en el

nodo de almacenamiento que usan coeficientes aleatorios; y/o

un número total de las porciones sustitutorias generadas por la pluralidad de nodos de almacenamiento (106) es igual a un parámetro de fiabilidad, en donde el parámetro de fiabilidad se basa en un número total de nodos de almacenamiento requeridos para que el archivo de datos siga siendo recuperable; y/o

cada uno de la pluralidad de nodos de almacenamiento (106) está configurado además para enviar información de comprobación *hash* al nodo de almacenamiento nuevo con el fin de verificar la porción sustitutoria; y/o

cada una de la pluralidad de porciones generadas a partir del archivo de datos comprende una combinación lineal de fragmentos del archivo de datos y un conjunto de coeficientes usados para generar la combinación; y/o

la pluralidad de porciones generadas a partir del archivo de datos se genera usando codificación de borrado (*erasure coding*) basada en fragmentos del archivo de datos; y/o

la porción sustitutoria generada se usa para reconstruir el archivo de datos o en donde la porción sustitutoria generada se usa para generar porciones sustitutorias subsiguientes para un nodo nuevo subsiguiente; y/o

la indicación se recibe desde un módulo de seguimiento (108) que monitoriza el estado de la pluralidad de nodos de almacenamiento (106) y realiza un seguimiento de posiciones de porciones.

9. Sistema según la reivindicación 8, en el que el nodo de almacenamiento nuevo sustituye a un nodo de almacenamiento fallido.

10. Sistema según la reivindicación 8, en el que la generación de una porción sustitutoria comprende:

multiplicar cada una de la pluralidad de porciones contenidas en el nodo de almacenamiento y un conjunto de coeficientes por un valor de escala aleatorio; y

combinar las porciones multiplicadas y el conjunto multiplicado de coeficientes.

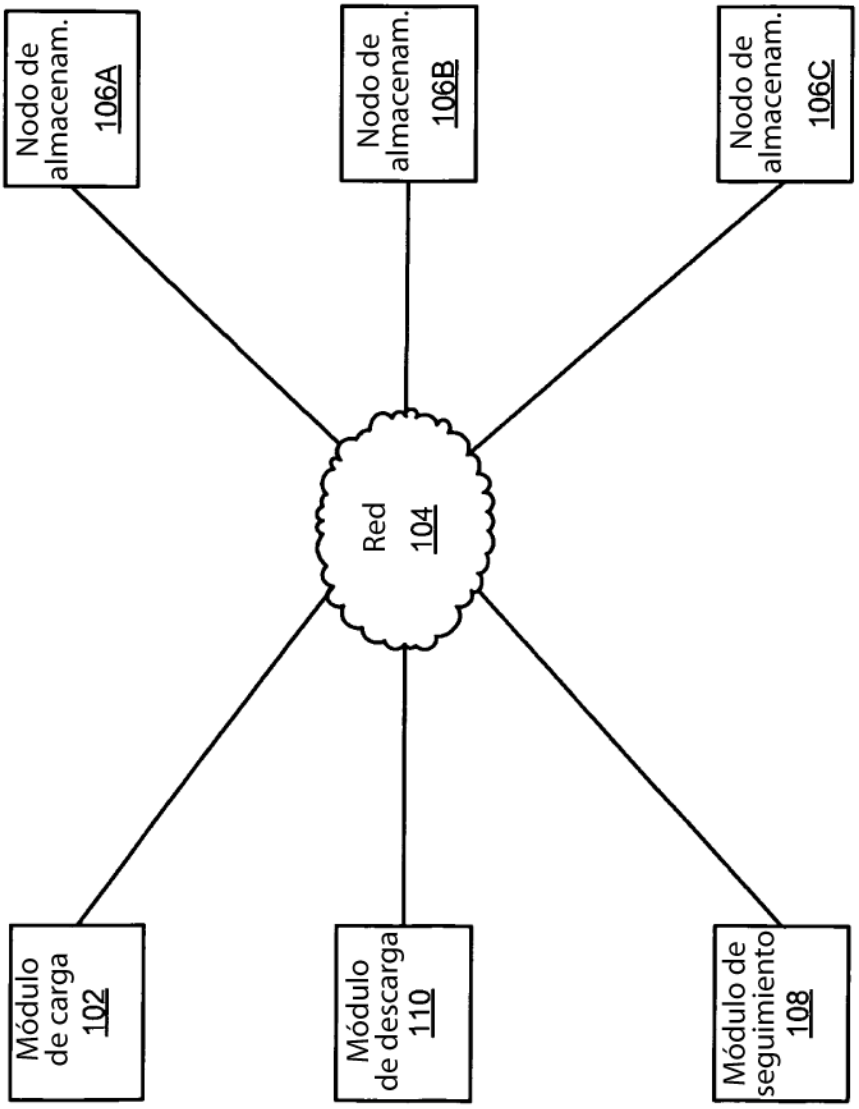
Nodo de almacenam. 106A
Nodo de almacenam. 106B
Nodo de almacenam. 106C
Red 104
Módulo de carga 102
Módulo de descarga 110
Módulo de seguimiento 108

FIG. 1

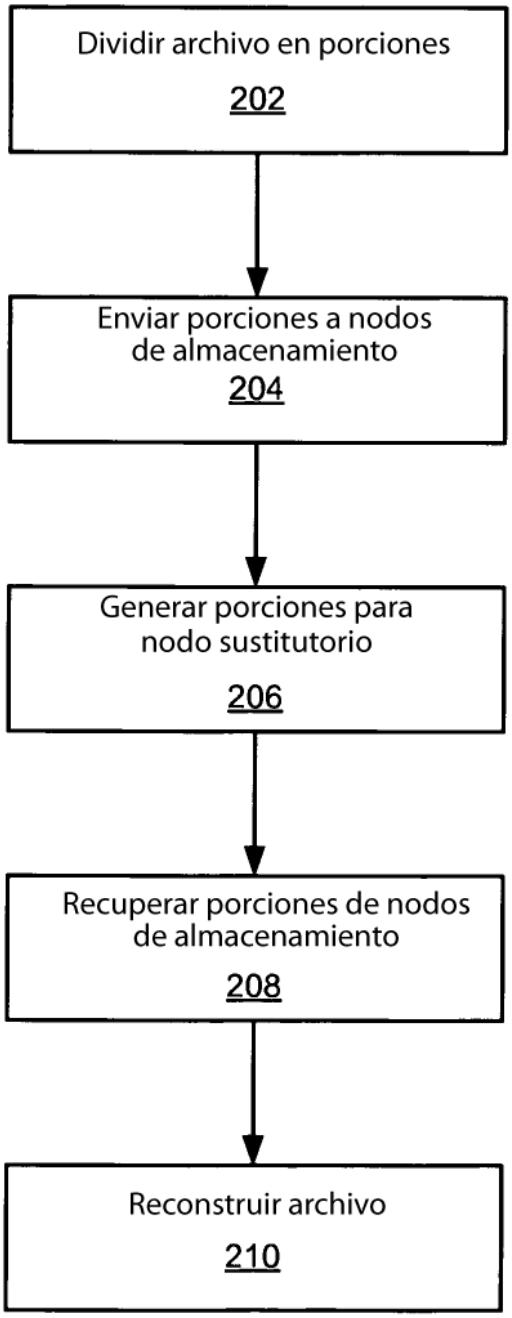
Dividir archivo en porciones
202
Enviar porciones a nodos de almacenamiento
204
Generar porciones para nodo sustitutorio
206
Recuperar porciones de nodos de almacenamiento
208
Reconstruir archivo
210

FIG. 2

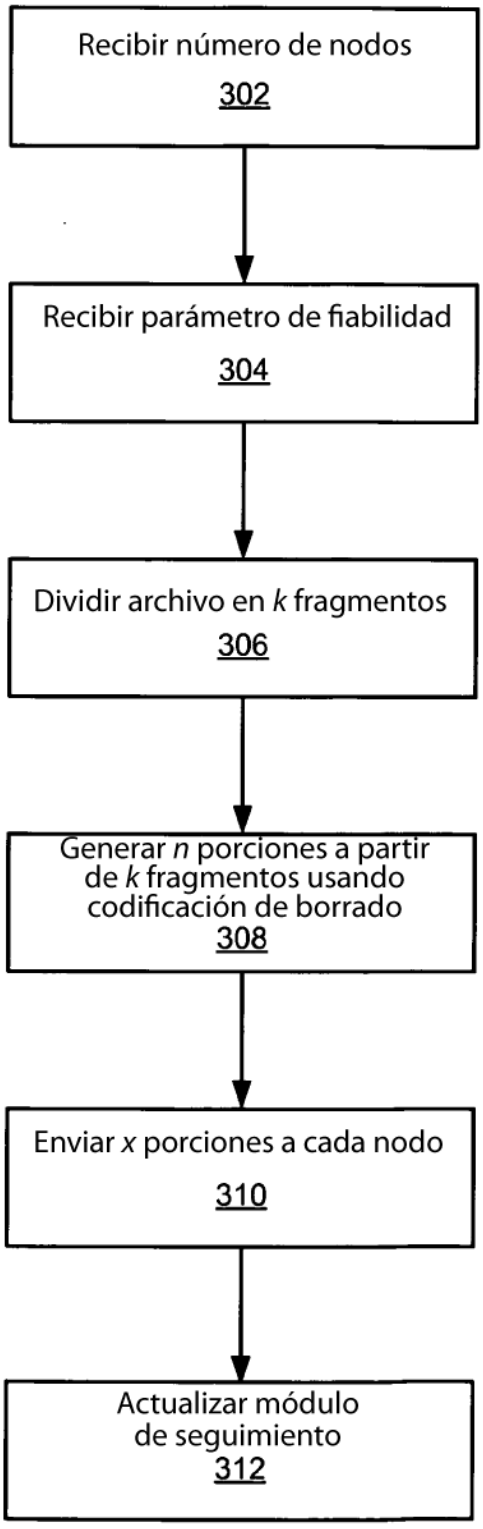
Recibir número de nodos
302
Recibir parámetro de fiabilidad
304
Dividir archivo en k fragmentos
306
Generar n porciones a partir de k fragmentos usando codificación de borrado
308
Enviar x porciones a cada nodo
310
Actualizar módulo de seguimiento
312

FIG. 3

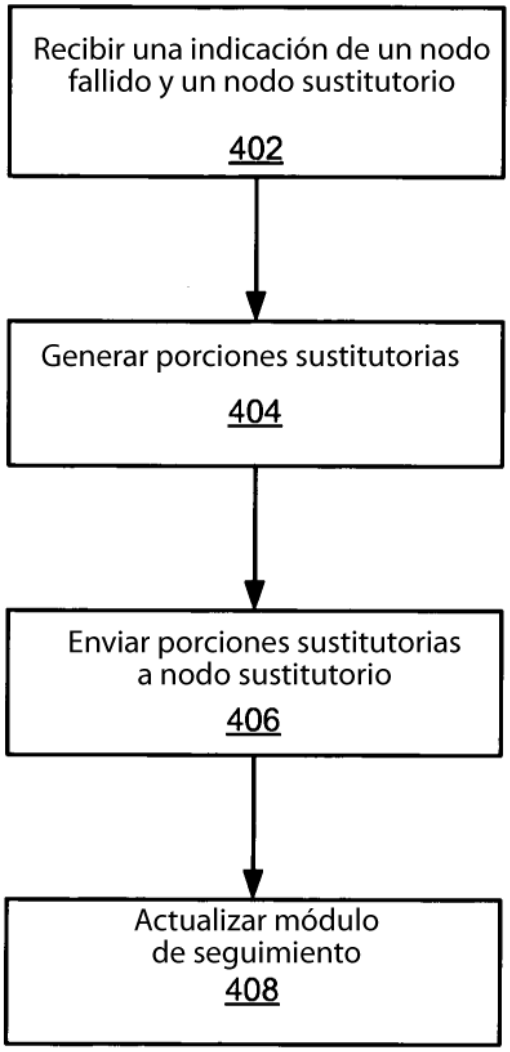
Recibir una indicación de un nodo fallido y un nodo sustitutorio
402
Generar porciones sustitutorias
404
Enviar porciones sustitutorias a nodo sustitutorio
406
Actualizar módulo de seguimiento
408

FIG. 4

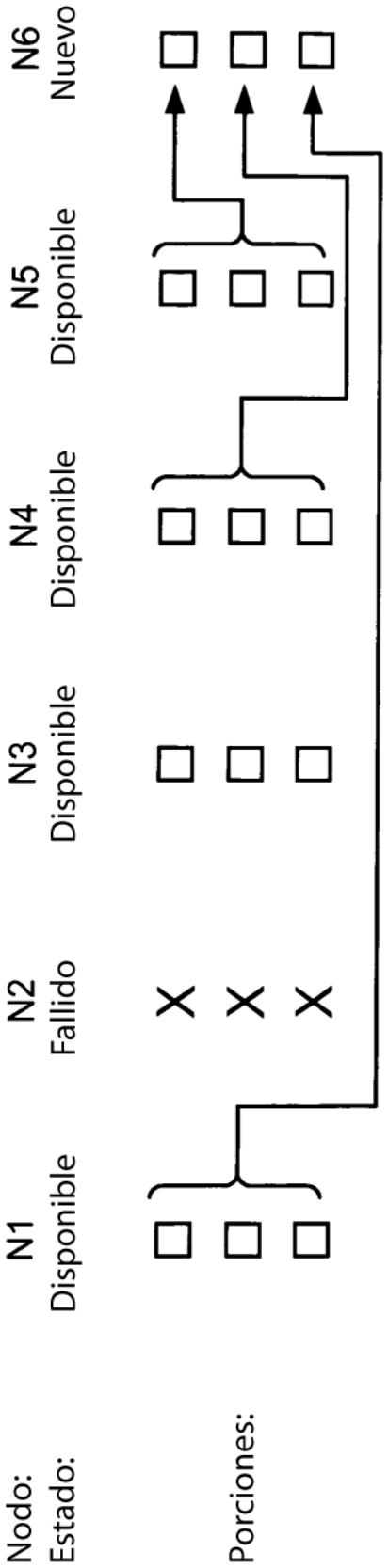
Nodo:
Estado:
N1
Disponible
N2
Fallido
N3
Disponible
N4
Disponible
N5
Disponible
N6
Nuevo
Porciones:

FIG. 5