(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 703 901**

(51) Int. CI.:

G06F 11/20 (2006.01)
G06F 17/30 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **29.08.2014 PCT/US2014/053404**

(87) Fecha y número de publicación internacional: **05.03.2015 WO15031755**

(96) Fecha de presentación y número de la solicitud europea: **29.08.2014 E 14839379 (6)**

(97) Fecha y número de publicación de la concesión europea: **03.10.2018 EP 3039549**

(54) Título: **Sistema de archivo distribuido mediante nodos de consenso**

(30) Prioridad:

29.08.2013 US 201314013948

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
13.03.2019

(73) Titular/es:

WANDISCO, INC. (100.0%)
Suite 270 Bishop Ranch 8 5000 Executive Parkway
San Ramon, CA 94583, US

(72) Inventor/es:

SHVACHKO, KONSTANTIN V.;
SUNDAR, JAGANE;
PARKIN, MICHAEL y
AAHLAD, YETURU

(74) Agente/Representante:

ELZABURU, S.L.P

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

Sistema de archivo distribuido mediante nodos de consenso

Antecedentes

El espacio de nombre de sistema de archivo distribuido Hadoop (HDFS) es una jerarquía de archivos y directorios. Los archivos y directorios se representan en el nodo de nombre por inodos. Los inodos registran atributos como permisos, tiempos de modificación y acceso, cuotas de espacio de nombre y espacio de disco. El contenido del archivo se divide en grandes bloques de datos (normalmente 128 MB) y cada bloque de datos del archivo es replicado independientemente en múltiples nodos de datos (normalmente tres). El nodo de nombre es el servicio de metadatos de HDFS que es responsable de las operaciones de espacio de nombre. El nodo de nombre mantiene el árbol de espacio de nombre y el mapeo de bloques de nodos de datos. Esto significa que el nodo de nombre rastrea la ubicación de los datos dentro de un clúster Hadoop y coordina el acceso del cliente al mismo. De manera convencional, cada clúster tiene un único nodo de nombre. El clúster puede tener miles de nodos de datos y decenas de miles de clientes HDFS por clúster, ya que cada nodo de datos puede ejecutar múltiples tareas de aplicación simultáneamente. Los inodos y la lista de bloques de datos que definen los metadatos del sistema de nombre se denominan la imagen. El nodo de nombre mantiene toda la imagen de espacio de nombre en la RAM. El registro constante de la imagen se almacena en el sistema de archivo nativo local del nodo de nombre como un punto de control además de un diario que representa actualizaciones del espacio de nombre llevado a cabo desde la realización del punto de control.

Un sistema distribuido está compuesto de diferentes componentes llamados nodos. Para mantener la consistencia del sistema, puede resultar necesario coordinar varios eventos distribuidos entre los nodos. La forma más sencilla de coordinar un evento particular que debe ser aprendido sistemáticamente por todos los nodos es seleccionando un único maestro designado y registrando dicho evento en el maestro, de modo que otros nodos puedan informarse del evento a partir del maestro. Este enfoque, aunque es simple, carece de fiabilidad, ya que el fracaso del maestro único paraliza el progreso del sistema completo. Al reconocer esto y tal como se muestra en la Fig. 1, las implementaciones HDFS convencionales usan un nodo de nombre activo 102 al que se accede durante operaciones normales y una copia de seguridad llamada nodo de nombre en espera 104 que se utiliza como conmutación por error en caso de fallo del nodo de nombre activo 102.

Como se muestra en la Fig. 1, un clúster HDFS convencional funciona de la siguiente manera. Cuando se solicita una actualización del espacio de nombre, como cuando un cliente HDFS emite una llamada a procedimiento remoto (RPC) para crear, por ejemplo, un archivo o un directorio, el nodo de nombre activo 102, como se muestra en la Fig. 1:

1. recibe la solicitud (por ejemplo, RPC) de un cliente;

2. aplica inmediatamente la actualización a su estado de memoria;

3. escribe la actualización como una transacción de diario en el almacenamiento persistente compartido 106 (como un almacenamiento conectado a la red (NAS) que comprende uno o más discos duros) y devuelve al cliente una notificación de confirmación.

El nodo de nombre en espera 104 debe actualizar ahora su propio estado para mantener la consistencia con el nodo de nombre activo 102. Con este fin, el nodo de nombre en espera 104

4. lee la transacción de diario del diario de transacción 106, y

5. actualiza su propio estado.

Esto, sin embargo, se cree que es una solución insuficiente. Por ejemplo, en este esquema, el propio diario de transacción 106 se convierte en el único punto de fallo. De hecho, una vez que se cumpla el diario de transacciones 106, el nodo de nombre en espera 104 no puede seguir asumiendo el mismo estado que el nodo de nombre activo 102 y deja de ser posible la conmutación por error desde el nodo de nombre activo al nodo en espera.

Además, en las soluciones de Hadoop que admiten solo un nodo de nombre por clúster, los servidores en espera, como se ha señalado antes, normalmente se mantienen sincronizados a través de dispositivos de almacenamiento conectado a la red (NAS). Si el nodo de nombre activo falla y el en espera tiene que asumir el control, existe la posibilidad de pérdida de datos si un cambio escrito en el nodo de nombre activo aún no se ha escrito en el NAS. El error del administrador durante la conmutación por error puede provocar una mayor pérdida de datos. Además, si se produce un fallo de red debido al cual el servidor activo no puede comunicarse con el servidor en espera, pero puede comunicarse con otras máquinas en el clúster, y el servidor en espera asume erróneamente que el servidor activo está muerto y asume el papel activo, puede ocurrir una condición de red patológica conocida como “cerebro dividido”, en la que dos nodos creen que son el nodo de nombre activo, cuya condición puede conducir a la corrupción de datos.

En el documento US2013/0111261 se describe un sistema de archivos de la técnica anterior. El procedimiento del documento US2013/0111261 requiere la elección de un maestro actual entre candidatos de un maestro m. La elección se realiza mediante un voto mayoritario basado en el *quorum* (también denominado en la técnica como "elección de liderazgo basada en el voto mayoritario basado en el *quorum*" o consenso distribuido del voto mayoritario basado en el *quorum*") entre las máquinas de *quorum*. El maestro actual elegido de esta manera recibe asignaciones por tiempo limitado actual de cierta duración, preferiblemente del orden de unos pocos segundos a unas pocas decenas de segundos. La asignación se mide mediante un reloj local que pertenece al maestro actual. Mientras mantiene la asignación o, de manera diferente, hasta el vencimiento de la asignación, el maestro actual ejecuta uno o más servicios solicitados por uno o más clientes de la red.

Según la invención, se proporciona un clúster de nodos que comprende dispositivos informáticos según la reivindicación 1 y un procedimiento implementado por ordenador para implementar un sistema de archivo distribuido según la reivindicación 8.

Breve descripción de los dibujos

La Fig. 1 es un diagrama de una implementación HDFS convencional.

La Fig. 2 es un diagrama de un sistema de archivo distribuido y aspectos de la actualización de nodos de nombre de consenso según una realización.

La Fig. 3 es un diagrama que ilustra aspectos de un procedimiento de replicación y generación de bloques en un sistema de archivo distribuido, según una realización.

La Fig. 4 es un diagrama que ilustra aspectos adicionales de la replicación de bloques según una realización.

La Fig. 5 es un diagrama que ilustra otros aspectos adicionales de la replicación de bloques según una realización.

La Fig. 6 es un diagrama que ilustra una manera para hacer identificadores de bloque únicos a través de nodos de nombre de consenso, según una realización.

La Fig. 7 es un diagrama de flujo de un procedimiento implementado por ordenador para implementar un sistema de archivo distribuido que comprende una pluralidad de nodos de datos configurados para almacenar bloques de datos de archivos de clientes, según una realización.

Descripción detallada

Las funciones de los proponentes (procesos que hacen propuestas a los miembros), los aceptadores (procesos que votan sobre si una propuesta debe ser acordada por los miembros) y los aprendices (procesos en los miembros que conocen los acuerdos que se han realizado) se definen, por ejemplo, en la implementación del algoritmo de Paxos descrito en Lamport, L.: The Part-Time Parliament, ACM Transactions on Computer Systems 16, 2 (mayo de 1998), 133-169. Según un ejemplo, pueden configurarse múltiples nodos, llamados aceptadores para almacenar eventos. Los eventos pueden presentarse como propuestas a un *quorum* de aceptadores. Mediante un protocolo interno, los aceptadores pueden acordar el orden del evento en una secuencia global de eventos. Una vez que se llega al acuerdo, los aceptadores dan a conocer a los aprendices los eventos en un orden consistente para todos los aprendices en el sistema. Por lo tanto, un motor de coordinación (como se muestra en 208 en la Fig. 2) puede comprender un conjunto de aceptadores junto con un protocolo que permite a los aceptadores acordar el orden de los eventos enviados al motor por múltiples proponentes. Para lograr confiabilidad, disponibilidad y escalabilidad, se pueden proporcionar múltiples nodos de nombre activos simultáneamente, según un ejemplo, replicando el estado del espacio de nombre en múltiples nodos con el requisito de que el estado de los nodos en los que se replica el espacio de nombre siga siendo consistente entre dichos nodos.

Esta consistencia entre nodos de nombre puede garantizarse por el motor de coordinación que puede configurarse para aceptar propuestas de actualización del espacio de nombre, optimizar las propuestas en una secuencia global de actualizaciones y solo entonces permitir que los nodos de nombre conozcan y apliquen las actualizaciones a sus estados individuales en el orden acordado. En esta memoria, la "consistencia" significa equivalencia de una copia, como se detalla en Bernstein y col., "Concurrency Control & Recovery in Database Systems", publicado por Addison Wesley, 1987, capítulos 6, 7 y 8. Dado que los nodos de nombre comienzan desde el mismo estado y aplican las mismas actualizaciones determinísticas en el mismo orden determinístico, sus respectivos estados son y siguen siendo consistentes.

Según un ejemplo, por lo tanto, el espacio de nombre puede replicarse en múltiples nodos de nombre, siempre y cuando

a) cada nodo pueda modificar su réplica de espacio de nombre, y

b) las actualizaciones de una réplica de espacio de nombre deban propagarse a las réplicas de espacio de nombre en otros nodos, de modo que las réplicas de espacio de nombre sigan siendo consistentes entre sí a través de los nodos.

Por lo tanto, un ejemplo elimina el único punto de fallo más problemático que afecta la disponibilidad, el nodo de nombre simple. De manera convencional, si el nodo de nombre simple ya no está disponible, el clúster Hadoop se desactiva y se requieren procedimientos de conmutación por error complejos (tales como la conmutación desde un nodo de nombre previamente activo a un nodo de nombre en espera) para restaurar el acceso. Para resolver este único punto de fallo potencial, una realización habilita múltiples servidores de nodos de nombre activos (en esta memoria con varias denominaciones como nodo de consenso o nodos C) para que actúen como pares, sincronizados constantemente y proporcionando simultáneamente acceso al cliente, incluyendo el acceso a aplicaciones en lote mediante MapReduce y a aplicaciones en tiempo real mediante Hbase. Según un ejemplo, cuando un servidor de nodo de nombre falla o es deshabilitado por un usuario para su mantenimiento o por cualquier otra razón, otros servidores de nodo de nombre activos por pares siempre están disponibles, lo que significa que no hay interrupción en el acceso a la lectura y escritura de los metadatos HDFS. En cuanto este servidor vuelve a estar en línea, su nodo de nombre se recupera automáticamente, se informa de cualquier cambio nuevo del espacio de nombre que pudiera haber ocurrido en este intermedio y sincroniza su espacio de nombre para hacer coincidir el espacio de nombre de todos los demás nodos de nombre en el clúster. Será consistente con las otras réplicas ya que reconoce los cambios en el mismo orden determinístico que los otros nodos han reconocido los cambios.

La Fig. 2 es un diagrama de un sistema de archivo distribuido y aspectos de actualización del nodo de consenso según una realización. Según un ejemplo, en vez de un único nodo de nombre activo y un nodo de nombre en espera, un clúster puede comprender una (preferiblemente impar) pluralidad (por ejemplo, 3, 5, 7…) de nodos de nombre que son coordinados por un motor de coordinación 208. Como se ha señalado anteriormente, en esta memoria, un nodo de nombre coordinado recibe el nombre de nodo de consenso o, en lo sucesivo, nodo C. Como se muestra en la Fig. 2, una realización puede comprender tres nodos C 202, 204, 206, acoplado cada uno al motor de coordinación 208. Según un ejemplo, el motor de coordinación 208 puede configurarse como un agente en cada nodo, coordinándose entre sí los agentes a través de una red. Sin embargo, para facilitar su consulta y descripción, el motor de coordinación 208 se muestra en las Figs. 2 y 4 como una entidad separada e independiente. Según un ejemplo, las actualizaciones del espacio de nombre iniciadas en una instancia del nodo de nombre 202, 204 o 206 se propagan a otras instancias de una manera consistente por medio del motor de coordinación 208. De este modo, los clientes acceden a un espacio de nombre consistente a través de todas las instancias del nodo de nombre. Los procedimientos de replicación descritos en esta memoria proporcionan un modelo activo-activo de alta disponibilidad para un sistema de archivo distribuido, tal como HDFS, en el que las solicitudes de metadatos (lectura o escritura) pueden ser de carga balanceada entre múltiples instancias del nodo de nombre.

El motor de coordinación 208 puede configurarse para determinar el orden global de las actualizaciones del espacio de nombre. Dado que todas las instancias del espacio de nombre comienzan en el mismo estado y que todos los nodos son susceptibles de aplicar las actualizaciones en el mismo orden determinístico (pero no necesariamente, según los ejemplos, al mismo tiempo), el estado de las múltiples instancias del espacio de nombre seguirá siendo consistente (o impulsado a la consistencia) a través de los nodos.

Según un ejemplo, y como se muestra en la Fig. 2, las actualizaciones consistentes a las múltiples réplicas del nodo C 202, 204, 206 pueden realizarse de la siguiente manera. Como se muestra en (1), uno de estos nodos C (en este caso, el nodo C 202) recibe una solicitud para actualizar el espacio de nombre de un cliente. Dicha actualización puede comprender una RPC, identificada en la Fig. 2 como RPC 3. De manera similar, en este ejemplo, el nodo C 204 recibe la RPC 1 y el nodo C 206 recibe la RPC 2. Las RPC pueden comprender una solicitud para añadir bloques de datos a un archivo, crear un archivo o crear un directorio, por ejemplo. Según un ejemplo, mientras el nodo C 202 actualiza inmediatamente su estado con el evento (por ejemplo, leer, escribir, eliminar, etc.) encapsulado dentro de la RPC 3, el nodo C 204 actualiza inmediatamente su estado con el evento encapsulado dentro de la RPC 1 recibida y el nodo C 206 actualiza inmediatamente su estado con el evento encapsulado dentro de la RPC 2 recibida y posteriormente se propagan los espacios de nombre actualizados a uno de los otros nodos C 202, 204, 206. En cambio, estas actualizaciones separadas de las réplicas de espacio de nombre en los nodos C pasan como propuestas al motor de coordinación 208, que luego emite los acuerdos correspondientes a los nodos C 202, 204, 206. De hecho, según un ejemplo, el mecanismo a través del que se mantiene la consistencia de las réplicas de espacio de nombre almacenadas por los nodos C 202, 204, 206 es la emisión de propuestas y la recepción de acuerdos desde el motor de coordinación 208. Esto significa que, como se muestra en la Fig. 2, en respuesta a la recepción de la RPC 3, el nodo C 202 puede emitir una propuesta Prop3 al motor de coordinación 208 como se muestra en (2). De manera similar, en respuesta a la recepción de la RPC 1, el nodo C 204 puede emitir una propuesta Prop1 al motor de coordinación 208 como se muestra en (2) y en respuesta a la recepción de la RPC 2, el nodo C 206 puede emitir una propuesta Prop2 al motor de coordinación 208 como también se muestra en (2). El motor de coordinación 208, según un ejemplo, ordena luego las propuestas que recibe como se muestra en (3) y suministra de retorno los acuerdos ordenados (en este caso, ordenados como ACU3, ACU1 y ACU2) a los nodos C 202, 204, 206 como se muestra en (4). Los nodos C 202, 204 y 206, a partir de la recepción de la secuencia ordenada de acuerdos ACU3, ACU1 y ACU2, aplica estos acuerdos a sus respectivos estados de memoria en ese orden determinístico, de modo que las réplicas de espacio de nombre puedan mantenerse de forma consistente a través de los nodos C 202, 204, 206. De esta forma, el estado de los nodos C 202, 204, 206 puede actualizarse asincrónicamente, como se muestra en (5) sin pérdida de consistencia. Estas actualizaciones pueden guardarse luego (aunque no necesariamente) como transacciones de diario en el almacenamiento persistente local respectivo 210, 212, 214 que pueden (aunque no necesariamente, como se indica con líneas discontinuas en 210, 212 y 214)

acoplarse o estar accesibles a los nodos C 202, 204, 206. Seguidamente, pueden devolverse notificaciones a los clientes del nodo C 202, 204, 206, informando a los clientes sobre el éxito de la actualización.

Por lo tanto, según un ejemplo, los nodos C 202, 204, 206 no aplican directamente las solicitudes del cliente a sus respectivos estados, sino más bien las redirigen como propuestas al motor de coordinación 208 para su orden. Las actualizaciones de los nodos C se emiten luego desde el motor de coordinación 208 como un conjunto ordenado de acuerdos. Esto garantiza la actualización de cada nodo C 202, 204, 206 cuando el cliente solicita el cambio de uno de ellos y la aplicación transparente y consistente de las actualizaciones de todos los nodos C en el clúster.

Por ejemplo, si un cliente crea un directorio a través del nodo C 202, y luego trata de enumerar el directorio recién creado a través del nodo C 204, el nodo C 204 debe retornar una excepción de “archivo no encontrado”. De manera similar, un cliente puede leer una cantidad diferente de *bits* del último bloque de datos de un archivo que se encuentra en construcción debido a que las réplicas del mismo bloque en diferentes nodos de datos tienen diferentes longitudes mientras los datos están en transición de un nodo de datos a otro, como se detalla a continuación con relación a la Fig. 3. Esto se conoce como un problema de “lectura obsoleta”.

Por lo tanto, un papel significativo del motor de coordinación 208, según una realización, es procesar las propuestas de modificación de estado de espacio de nombre de todos los nodos C y transformarlas en una secuencia ordenada global de acuerdos. Los nodos C pueden aplicarse después a los acuerdos desde esa secuencia ordenada como actualizaciones de su estado. Los acuerdos pueden ordenarse, según una realización, según un número de secuencia global (GSN), que puede configurarse como un único número monótono creciente. El GSN puede configurarse de otro modo, como pueden reconocer los expertos en la técnica. El GSN puede utilizarse posteriormente para comparar el progreso de diferentes nodos C con respecto a la actualización del estado del espacio de nombre y al mantenimiento de la consistencia en el estado del espacio de nombre a través de los nodos C. Por ejemplo, si el nodo C 202 acaba de procesar un acuerdo enumerado GSN1, que es menor que GSN2 que acaba de ser procesado por el nodo C 204, entonces el nodo C 202 tiene un estado de espacio de nombre más reciente que el nodo C 204.

Según un ejemplo, con cada operación, los clientes se informan sobre el último GSN procesado en el nodo C al que está conectado actualmente el cliente. Posteriormente, si el cliente cambia a otro nodo C, debería primero, según una realización, esperar (si es necesario) hasta que el nuevo nodo C recupere el último GSN que el cliente conoce (es decir, el GSN que el cliente recibió a partir del nodo C al que accedió anteriormente) antes de emitir una RPC que comprende un comando de acceso de datos. Esto evitará el problema de la lectura obsoleta.

Según un ejemplo, solo se requiere la coordinación por el motor de coordinación 208 de las operaciones que actualizan el estado del espacio de nombre. Esto significa que la mayoría (pero no todas, según una realización detallada a continuación) de las solicitudes de lectura no alteran el estado del espacio de nombre. Debe tenerse en cuenta que, según una realización, el motor de coordinación 208 no garantiza que todos los nodos C 202, 204, 206 tengan el mismo estado en cualquier momento dado. El motor de coordinación 208 garantiza más bien que cada nodo C 202, 204, 206 se informe eventualmente sobre cada actualización en el mismo orden que todos los demás nodos C, y que los clientes sean capaces de ver esta información. De este modo, el motor de coordinación 208 está configurado para generar una secuencia global ordenada de eventos que se suministra de manera similar a todos los nodos C 202, 204, 206.

Según un ejemplo, pueden realizarse actualizaciones de diario al almacenamiento persistente local 210, 212, 214. Sin embargo, la consistencia de los nodos C 202, 204, 206 no depende de dichas actualizaciones de diario y cada uno de los almacenamientos persistentes (si existen), según un ejemplo, tienen carácter local con respecto a un nodo C y no se comparten a través de los nodos C. De manera similar, el mantenimiento de la consistencia del estado de espacio de nombre a través de los nodos C 202, 204, 206 no depende de compartir otros recursos, tales como recursos de memoria o procesadores.

No hay un nodo C preferido (maestro o distinguido de otra manera), según los ejemplos. De hecho, si uno o más servidores de nodo C falla o se deshabilita para su mantenimiento (o por cualquier otra razón), siempre hay otros servidores de nodo C disponibles para servir a los clientes sin ninguna interrupción de acceso. Según un ejemplo, en cuanto el servidor vuelve a estar en línea, vuelve a sincronizarse con los otros servidores de nodo C automáticamente, como se describe a continuación. Dicha sincronización puede comprender el conocimiento de todos los acuerdos que fueron emitidos por el motor de coordinación 208 desde que se desactivó o inhabilitó el nodo C. Se elimina tanto la condición de cerebro dividido como la pérdida de datos, ya que todos los nodos C están activos y se mantienen en o se someten al sincronismo, proporcionando así una copia de seguridad en caliente por defecto. Tanto la conmutación por error como la recuperación son inmediatas y automáticas, lo cual elimina además la necesidad de una intervención manual y el riesgo de un error por parte del administrador. Además, ninguno de los nodos C 202, 204, 206 está configurado como nodo de nombre en espera pasivo. De hecho, según una realización, todos los servidores de nodo C en el clúster están configurados para soportar solicitudes de clientes simultáneamente. En consecuencia, esto hace posible que el clúster sea capaz de soportar servidores de nodo C adicionales, sin sacrificar el rendimiento a medida que la carga de trabajo aumenta. Según una realización, no hay servidores en espera pasivos y se eliminan completamente las vulnerabilidades e impedimentos de un único servidor de nodo de nombre activo. Además, al distribuirse las solicitudes de cliente a través de múltiples nodos C 202, 204,

206 se distribuye sustancialmente la carga de procesamiento y el tráfico sobre todos los nodos C disponibles. También se puede realizar una carga balanceada activa a través de los nodos C 202, 204, 206, en comparación con el paradigma de nodo de nombre activo/ en espera, en el que todas las solicitudes de los clientes son atendidas por un único nodo de nombre.

La Fig. 3 es un diagrama que ilustra aspectos de un procedimiento de replicación y generación de bloque en un sistema de archivo distribuido, según un ejemplo. En 350, la Fig. 3 muestra un archivo que debe almacenarse en HDFS. Según un ejemplo, la unidad de almacenamiento podría ser considerada un bloque y el tamaño del bloque podría ser considerablemente grande. Por ejemplo, el tamaño del bloque podría ser 128 MB de almacenamiento físico. También puede implementarse fácilmente otro tamaño de bloque. En la Fig. 3 se muestra el archivo 350 que comprende una pluralidad de bloques de datos de 128 MB. El tamaño de bloque no tiene que ser necesariamente 128 MB. Según un ejemplo, cada bloque de datos de un archivo puede replicarse (es decir, almacenarse de manera idéntica) en una pluralidad de nodos de datos. Dichos nodos de datos se muestran en 302, 304 y 306 y están configurados para acoplarse a uno o más nodos C, tal como el nodo C 202. Según un ejemplo, cada nodo de datos puede configurarse para comunicarse con cada uno de los nodos C en el clúster. Los bloques de datos de archivos pueden almacenarse en un mayor número de nodo de datos, tales como en 5 o 7 nodos de datos. Al almacenarse cada bloque de datos en múltiples nodos de datos, se proporciona fiabilidad de los datos gracias a la redundancia.

Como se muestra en la Fig. 2, un cliente envía un mensaje (por ejemplo, una RPC) al nodo C 202, indicando la intención del cliente de crear un archivo y escribir un bloque de datos en el archivo. El nodo C 202, según un ejemplo, puede seleccionar seguidamente múltiples nodos de datos (tres en esta implementación ejemplar) 302, 304 y 306 en los que se replicará el bloque de datos del archivo recién creado y así lo informa al cliente. El cliente puede luego, según un ejemplo, comenzar a transmitir (o, por el contrario, enviar) datos a uno de los tres nodos de datos 302, 304 y 306 seleccionados. Dicha transmisión puede realizarse mediante el envío en serie al nodo de datos seleccionado (nodo de datos 302, por ejemplo) de pequeños fragmentos de cada bloque de datos. Por ejemplo, el cliente puede enviar al nodo de datos 302 un flujo en serie de fragmentos de 64 KB del primer bloque de datos del archivo, hasta que el primer bloque de datos del archivo se haya transmitido con éxito al nodo de datos 302. El protocolo de intercambio entre el cliente y el nodo de datos 302 seleccionado puede garantizar que cada bloque de datos sea recibido y almacenado correctamente por el nodo de datos 302 seleccionado. Los fragmentos de datos enviados al primer nodo de datos 302 también pueden comprender una indicación del segundo nodo de datos 304 al que deben enviarse los bloques de datos del archivo del cliente. Según un ejemplo, en lugar de que el cliente envíe los bloques de datos directamente a los tres (o más) nodos de datos seleccionados por el nodo C 202 para recibir réplicas de los bloques de datos del archivo, el primer nodo de datos 302 que acaba de recibir un fragmento de datos del bloque puede por sí solo enviar luego el fragmento de datos recibido al próximo (por ejemplo, al nodo de datos 304) de los tres nodos de datos para recibir los bloques de datos del archivo. De manera similar, después de que el nodo de datos 304 haya recibido el fragmento de datos enviado al mismo por el nodo de datos 302, este puede enviar después el fragmento de datos al último de los tres nodos de datos seleccionados por el nodo C 202 para recibir réplicas de los bloques de datos constituyentes del archivo del cliente. De este modo, se crea un canal de fragmentos de datos, en el que un primer nodo de datos seleccionado por el nodo C reenvía fragmentos de datos al segundo nodo de datos seleccionado por el nodo C y en el que el segundo nodo de datos reenvía fragmentos de datos recibidos al tercer nodo de datos seleccionado por el nodo C para recibir réplicas del bloque de datos del archivo (y así sucesivamente, si más de tres nodos de datos deben recibir el bloque del archivo).

Según una realización, el nodo C no asume que los nodos de datos que ha seleccionado como receptores de los bloques de datos constituyentes del archivo del cliente hayan, de hecho, recibido y almacenado con éxito los bloques de datos. En cambio, según un ejemplo, una vez que tienen en su poder uno o más bloques de datos del archivo del cliente, los nodos de datos 302, 304, 306 pueden informar al nodo C 202 que tienen almacenada ahora una réplica del bloque de datos que se les ha enviado directamente por el cliente o por otro nodo de datos, como se muestra en la Fig. 3. Al menos uno (y según un ejemplo, cada uno) de los nodos de datos puede emitir periódicamente un mensaje de “señal de monitorización” a los nodos C, cuyo mensaje de señal de monitorización puede configurarse para informar a los nodos C de que el nodo de datos que se está emitiendo todavía está activo y en buen estado (es decir, capaz de atender las solicitudes de acceso de datos de los clientes). Los nodos de datos pueden, según un ejemplo, informar la recepción exitosa y el almacenamiento de uno o más bloques de datos del archivo del cliente como otro mensaje al nodo C. En la situación ejemplar representada en la Fig. 3, los nodos de datos 302, 304, 306 pueden informar al nodo C 202 que han recibido y almacenado con éxito uno o más bloques de datos del archivo del cliente en el nodo C 202.

Los nodos de datos pueden fallar. Si dicho fallo es ocasionado por una interrupción en el canal de comunicación entre el nodo de datos y el nodo C, un fallo de un servidor de archivo o un fallo del almacenamiento físico subyacente (o cualquier otro fallo), dicho fallo significa que los bloques de datos pueden no estar disponibles, al menos desde el nodo de datos fallido. En el ejemplo mostrado en la Fig. 4, el nodo de datos 306 ha fallado. Según un ejemplo, los nodos C 202, 204, 206 pueden no estar informados inmediatamente de este estado de cambio del nodo de datos 306. En cambio, se puede utilizar el mecanismo de mensaje de señal de monitorización descrito anteriormente para conseguir mantener informados a los nodos C del estado casi próximo (al igual que la última señal de monitorización) de cada nodo de datos. Esto significa que, según un ejemplo, el fallo de los nodos C al recibir un mensaje de señal de monitorización dentro de un periodo de tiempo determinado es interpretado, por los nodos C, como un fallo del nodo de datos que no envía señales de monitorización. Dicho periodo de tiempo

predeterminado puede establecerse, por ejemplo, en un periodo de tiempo superior al intervalo esperado entre mensajes de señal de monitorización de funcionamiento desde cualquier nodo de datos individual.

En el ejemplo de la Fig. 4, el nodo de datos 306 no ha podido enviar un mensaje de señal de monitorización ("HB" en la Fig. 3) dentro del intervalo de tiempo predeterminado desde su última señal de monitorización y puede considerarse, por tanto, su fallo y que los bloques de datos almacenados están, al menos por ahora, inaccesibles. A su vez, esto significa que solo los nodos de datos 302 y 304 almacenan los bloques de datos de diferentes archivos. Según un ejemplo, los nodos C pueden mantener una lista de nodos de datos que se encuentran actualmente activos y, según una realización, listos para aceptar nuevos bloques de datos y/ o solicitudes de acceso de datos de servicio. Dicha lista puede considerarse como una lista "activa". En caso de fallo en la recepción de un mensaje de señal de monitorización esperado de un nodo de datos, tal como el nodo de datos 306 en la Fig. 4, puede considerarse que el nodo de datos ha fallado y los nodos C pueden eliminar el nodo de datos fallido de la lista activa. Según un ejemplo, la lista activa puede ser dicha lista desde la que el nodo C, al recibir una solicitud de un cliente para crear un bloque, puede seleccionar los (por ejemplo) tres nodos C en los que se almacenará el bloque de datos del archivo que se creará. Como el nodo de datos 306 ha fallado, el nodo de datos 306 puede ser eliminado de la lista activa, haciendo que dicho nodo de datos, para cualquier fin, quede efectivamente inaccesible e inexistente, al menos desde el punto de vista de los nodos C.

Como los bloques de datos del archivo del cliente están subreplicados (por ejemplo, almacenados por debajo del número de nodos de datos predeterminado) debido al fallo del nodo de datos 306, el nodo C 202 puede seleccionar ahora, según un ejemplo, un nuevo nodo de datos en el que se puedan replicar los bloques de datos del archivo del cliente, para garantizar que un complemento total de tres nodos de datos almacene réplicas de los bloques de datos constituyentes del archivo. Según un ejemplo, el nodo C 202 puede consultar la lista activa y seleccionar, de la lista, un nuevo nodo de datos en el que se replicarán los bloques de datos del archivo del cliente para hacer que los complementos de nodos de datos que almacenan réplicas de los bloques de datos del archivo de la copia de seguridad del archivo del cliente sean hasta tres (o cuatro, cinco, etc., en función del factor de replicación asignado al archivo). En el ejemplo mostrado en la Fig. 4, el nodo C 202 ha seleccionado el nodo de datos 402 como el nodo de datos en el que se almacenarán también réplicas del bloque de datos, para solucionar la subreplicación del bloque de datos. Según un ejemplo, el nodo C 202 también puede seleccionar el nodo de datos 304 que enviará la réplica en su poder al nodo de datos 402 seleccionado. Como se muestra en 406 en la Fig. 4, el nodo de datos 304 seleccionado puede empezar luego a transmitir fragmentos de datos de la réplica de bloque o en cambio, enviar la réplica del bloque al nodo de datos 402 recién seleccionado. Cuando el nodo de datos 402 recién seleccionado recibe la réplica del bloque y cuando llega el momento en el que el nodo de datos 406 debe informar a los nodos C, puede informar que ahora almacena réplicas de los bloques recién recibidos. Los nodos C pueden cambiar el espacio de nombre para reflejar este cambio. Según un ejemplo, el nodo de datos receptor puede seleccionarse por el nodo C 202 de forma aleatoria. Según otras realizaciones, dicha selección puede realizarse según criterios de selección predeterminados.

Según un ejemplo, cada uno de los nodos C 202, 204, 206 está "informado" de cada uno de los nodos de datos 302, 304, 306, 402 y todos los demás (potencialmente miles) nodos de datos de quienes reciben periódicamente señales de monitorización. En caso de fallo de un nodo de datos, más de un nodo C podría decidir seleccionar un nodo de datos como un nodo de datos emisor y otro nodo de datos como el receptor de réplicas de bloque, para garantizar que los bloques no sean subreplicados. Esto podría traer consigo el hecho de que múltiples nodos C seleccionen múltiples nodos de datos de reemplazo para almacenar los bloques de datos almacenados anteriormente por un nodo de datos fallido. A su vez, dichas acciones paralelas podrían resultar en una sobrerreplicación de bloques (por ejemplo, replicación de más de las 3, 4, 5… instancias esperadas de los mismos). Dicha sobrerreplicación también puede producirse cuando, como se muestra en la Fig. 5, un nodo de datos fallido anteriormente o de lo contrario inaccesible vuelve a estar en línea. En la Fig. 5, se asume que el Nodo de datos 306 fallido anteriormente o inaccesible está ahora nuevamente operativo y accesible para los nodos C 202, 204, 206. En este estado, los bloques del archivo del cliente se encuentran ahora en cuatro nodos de datos; principalmente, los nodos originales 302, 304, el nodo de datos 402 añadido recientemente y el nodo de datos no operativo y accesible 306. Los bloques de datos del archivo del cliente se encuentran, por lo tanto, sobrerreplicados. Como ahora es conocido el estado de retorno en línea del nodo de datos 3 por todos los nodos C 202, 204, 206 (porque cada uno recibió una señal de monitorización del nodo de datos reactivado 306), resulta concebible que más de un nodo C 202, 204, 206 pueda seleccionar de manera independiente un nodo de datos desde el que se eliminen las réplicas de bloque del archivo del cliente. Esta selección independiente puede hacer que las réplicas de bloque del archivo del cliente vayan desde un estado sobrerreplicado a un estado subreplicado o en el peor de los casos incluso que se eliminen de todos los nodos de datos.

Para evitar dichas incidencias, según un ejemplo, las funciones de replicación de bloque deben reservarse a un único nodo C seleccionado o elegido en cualquier momento dado, el nodo C replicador de bloque. Dichas funciones de replicación de bloque, según una realización, pueden comprender la coordinación de la replicación de bloques (es decir, la indicación de los bloques que se copiarán entre nodos de datos) y las eliminaciones de bloque. La funcionalidad de la generación de bloques, según un ejemplo, no supone dichos riesgos inherentes de pérdida de datos o sobrerreplicaciones y pueden conferirse, por lo tanto, en cada nodo C del clúster. Por lo tanto, todos los nodos C pueden estar configurados para realizar funciones de gestión de bloques, según un ejemplo. Sin embargo, dichas funciones de gestión de bloques pueden dividirse en funciones de replicación y eliminación de bloques que,

según una realización, se reservan a un único nodo C seleccionado, y funciones de generación de bloques, que pueden conferirse en cada uno de los nodos C de un clúster. Esto se muestra en la Fig. 5, en la que el nodo C 202 se ha seleccionado como el único nodo C configurado con una función de replicador de bloque 410 para permitir que solo el nodo C 202 pueda autorizar la copia y/ o eliminación de los bloques de datos de los nodos de datos. En cambio, y como se muestra en la Fig. 5, cada uno de los nodos C 202, 204, 206 puede configurarse para realizar las funciones de generador de bloque 408, 412 y 414, respectivamente, permitiendo que cualquiera de los nodos C 202, 204 y 206 genere bloques o permite el almacenamiento de nuevos bloques de datos en los nodos de datos seleccionados que informan al mismo.

Cada nodo de datos, según un ejemplo, puede configurarse para enviar todas las comunicaciones a todos los nodos C en el clúster. Esto quiere decir que cada nodo de datos de trabajo, activo, puede configurarse para enviar señales de monitorización, informes de bloques y mensajes sobre réplicas recibidas o eliminadas, etc. de manera independiente a cada nodo C del clúster.

En la implementación de HDFS actual, los nodos de datos solo reconocen un nodo de nombre activo. A su vez, esto significa que los nodos de datos ignorarán cualquier comando de nodo de datos que venga de un nodo de nombre inactivo. De manera convencional, si un nodo de nombre inactivo afirma que es ahora el nodo de nombre activo y confirma dicho estado con un txId superior, el nodo de datos realizará un procedimiento de conmutación por error, conmutando a un nuevo nodo de nombre activo y aceptando solamente comandos de nodo de datos del nuevo nodo de nombre activo.

Para adaptar este procedimiento de operación en clústeres de nodo C según realizaciones, solo el nodo C que tiene funciones de replicador de bloque (es decir, el replicador de bloque actual) informa su estado como activo a los nodos de datos. Esto garantiza que solo el replicador de bloque tenga la habilidad de ordenar a los nodos de datos que repliquen o eliminen réplicas de bloque.

Las aplicaciones acceden al HDFS a través de los clientes de HDFS. De manera convencional, un cliente HDFS contactaría el único nodo de nombre activo de los metadatos del archivo y luego accedería a los datos directamente desde los nodos de datos. De hecho, en la implementación actual de HDFS, el cliente siempre habla con el único nodo de nombre activo. Si se habilita la alta disponibilidad (HA), el nodo de nombre activo puede conmutar por error a un nodo en espera. Cuando esto sucede, el cliente de HDFS se comunica con el nodo de nombre activo recientemente (el nodo en espera anterior) hasta y si ocurre otra conmutación por error. La conmutación por error es gestionada por una interfaz conectable (por ejemplo, proveedor de *proxy* de conmutación por error), que puede tener diferentes implementaciones.

Según ejemplos, sin embargo, todos los nodos C se activan todo el tiempo y pueden utilizarse igualmente para ofrecer información de espacio de nombre a los clientes. Según una realización, los clientes HDFS pueden configurarse para comunicarse con nodos C a través de una interfaz *proxy* llamada, por ejemplo, *proxy* de nodo C. Según un ejemplo, el *proxy* de nodo C puede configurarse para seleccionar aleatoriamente un nodo C y para abrir un puerto de comunicación para enviar las solicitudes de RPC del cliente a este nodo C seleccionado aleatoriamente. Luego el cliente solo envía solicitudes de RPC a este nodo C hasta que transcurra un límite de tiempo u ocurra un fallo. El límite de tiempo de comunicación puede ser configurable. Cuando el límite de tiempo de comunicación expira, el cliente puede conmutar a otro nodo C (seleccionado, por ejemplo, aleatoriamente por el *proxy* de nodo C), abrir un puerto de comunicación a este nuevo nodo C y enviar las solicitudes de RPC del cliente solo a este nuevo nodo C seleccionado aleatoriamente. Para fines de compensación de carga, por ejemplo, este límite de tiempo de comunicación puede establecerse a un valor bajo. De hecho, si el nodo C al que el cliente envía sus solicitudes de RPC está ocupado, el plazo de respuesta puede ser mayor que el valor inferior del límite de tiempo de comunicación, obligando al cliente a conmutar, a través del *proxy* de nodo C, el nodo C con el que se comunicará.

De hecho, la selección aleatoria de un nodo C por clientes HDFS permite la compensación de carga de múltiples clientes que se comunican con nodos C replicados. Cuando el *proxy* de nodo C ha seleccionado aleatoriamente el nodo C con el que el cliente se comunicará, el cliente puede “adherirse” a dicho nodo C hasta que, según un ejemplo, al nodo C seleccionado aleatoriamente se le agote el tiempo o falle. Esta “adherencia” al mismo nodo C reduce la posibilidad de lecturas obsoletas, descritas anteriormente, al caso de conmutación por error solamente. El *proxy* del *proxy* de nodo C puede configurarse para que no seleccione nodos C que se encuentran en modo seguro, lo que puede ocurrir cuando el nodo C se está reiniciando y no está todavía totalmente preparado para el servicio (por ejemplo, está informándose sobre los acuerdos que pudo haber ignorado durante su tiempo de inactividad).

El problema de la lectura obsoleta descrita anteriormente puede ilustrarse además a través de un ejemplo. Por ejemplo, si un cliente crea un directorio a través del nodo C1 y luego el mismo u otro cliente trata de listar el directorio recién creado a través del nodo C2, el nodo C2 puede estar detrás de dicho proceso informativo y puede devolver la excepción de archivo no encontrado porque no ha recibido o procesado todavía el acuerdo para crear el directorio. De manera similar, un cliente puede leer diferentes números de *bits* del último bloque de un archivo que se encuentra en construcción, ya que las réplicas del mismo bloque en diferentes nodos de datos pueden tener longitudes diferentes mientras los datos están en transición.

El problema de lectura obsoleta puede manifestarse en dos casos:

1. Un mismo cliente conmuta (debido al fallo, interrupción intencional o por razones de compensación de carga, por ejemplo) a un nuevo nodo C, que tiene un estado de espacio de nombre más antiguo y;

2. Un cliente modifica el espacio de nombre que debe ser visto por otros clientes.

El primer caso puede evitarse, según una realización, informando al *proxy* de nodo C de la interfaz de *proxy* el GSN del nodo C al que se conecta. Con cada operación, el cliente HDFS conoce el GSN en el nodo C. Cuando el cliente conmuta a otro nodo C (por ejemplo, debido a un fallo del nodo C, límite de tiempo o un apagado deliberado de dicho nodo C por cualquier razón, el cliente, a través del *proxy* de nodo C, debe elegir un nodo C con un GSN que no sea inferior al que ya ha visto o esperar hasta que el nuevo nodo C se actualice con el último GSN que el cliente recibió del nodo C anterior.

El segundo caso surge cuando se inicia un trabajo de MapReduce. En este caso, un cliente de MapReduce coloca los archivos de configuración de trabajo tal como trabajo.xml en el HDFS, que luego es leído por todas las tareas ejecutadas en el clúster. Si alguna tarea se conecta a un nodo C que no conoce los archivos de configuración de trabajo, la tarea fallará. De manera convencional, dicha restricción requiere coordinación externa entre los clientes. Sin embargo, la coordinación entre clientes es reemplazada, según una realización, por lecturas coordinadas.

Según un ejemplo, se puede realizar una lectura coordinada del mismo modo que las operaciones de modificación. Esto significa que un nodo C presenta una propuesta de lectura del archivo y realmente lo lee cuando el acuerdo correspondiente es recibido de retorno desde el motor de coordinación 208. Por lo tanto, los acuerdos de lectura, según un ejemplo, pueden ejecutarse en la misma secuencia global como acuerdos de modificación de espacio de nombre, garantizando así que las lecturas coordinadas nunca sean obsoletas. Según un ejemplo, no se necesita utilizar las lecturas coordinadas para todas las lecturas, ya que esto podría aumentar innecesariamente la carga computacional en el motor de coordinación 208 y podría ralentizar la eficacia de la lectura en el clúster. En consecuencia, según un ejemplo, solo archivos seleccionados, tal como trabajo.xml pueden exponerse a lecturas coordinadas. Por consiguiente, según una realización, un conjunto de patrones de nombre de archivo puede definirse, por ejemplo, como un parámetro de configuración. Dichos patrones pueden reconocerse por los nodos C de un clúster. Cuando se definen dichos patrones de nombre de archivo, el nodo C hace corresponder los nombres de archivo que se leerán con los patrones de nombre de archivo, y si la correspondencia es positiva, el nodo C realiza una lectura coordinada de dicho archivo.

Si un cliente ha accedido una vez a un objeto en un nodo C particular, se requiere que el acceso no se realice a través de lecturas coordinadas para clientes posteriores. Según una realización, se puede identificar un archivo por el hecho de haberse accedido al mismo a través de solicitudes de RPC. De este modo, si un nodo C que ejecuta dicha solicitud ve que el archivo no ha sido identificado, dicho nodo C puede presentar una propuesta al motor de coordinación 208 y esperar a recibir el acuerdo correspondiente para realizar una lectura coordinada. Este acuerdo de lectura llega a todos los nodos C, que pueden identificar sus réplicas de archivo a las que se ha accedido. Todas las solicitudes de cliente posteriores para acceder al archivo identificado, según una realización, no necesitan ser de lectura coordinada. Por lo tanto, en el peor de los casos con tres nodos C en el clúster, no puede haber más de tres lecturas coordinadas por archivo, manteniendo a un alto nivel la eficacia de lectura.

Los nodos C también pueden fallar o deshabilitarse de manera intencional por razones de mantenimiento. Si un nodo C fallido es también el único nodo C que se ha dedicado con funciones de replicador de bloque (esto significa que ha sido elegido como el replicador de bloque), entonces el clúster puede abandonarse sin la capacidad de replicar o eliminar bloques de datos. Por lo tanto, según una realización, el nodo C que tiene la función de replicador de bloque como se muestra en 410 puede configurarse para enviar también señales de monitorización de replicador de bloque (BR HB), como se muestra en 416, al motor de coordinación 208. Mientras que el motor de coordinación 208 reciba BR HB periódicas 416 desde el nodo C seleccionado como incluyen las funciones de replicador de bloque 410, dicho nodo C puede continuar realizando dichas funciones de replicación de bloque. Sin embargo, en caso de que el motor de coordinación 208 falle en la recepción puntual de una o más BR HB desde el nodo C seleccionado como el replicador de bloque (410), las funciones de replicación de bloque serán asignadas a otro de los nodos C dentro del clúster. A su vez, el nodo C seleccionado de este modo puede emitir posteriormente BR HB periódicas (que se diferencian de las señales de monitorización HB emitidas por los nodos de datos) al motor de coordinación 208 y puede continuar desempeñando dicho papel hasta que el motor de coordinación 208 pueda recibir una o más BR HB, tras lo cual el proceso de selección de nodo C puede repetirse.

Según un ejemplo, con el fin de garantizar la singularidad del replicador de bloque 410 en el clúster, el nodo C que comprende el replicador de bloque 410 puede configurarse para emitir periódicamente una propuesta de replicador de bloque al motor de coordinación 208. A su vez, el motor de coordinación 208, tras la recepción de la propuesta de replicador de bloque, puede confirmar dicho nodo C como el seleccionado o elegido para cumplir las funciones de replicación de bloque, confirmando su misión de replicador de bloque a todos los nodos C en el clúster. Si un BR HB no es escuchado por los nodos C durante un periodo de tiempo configurable, otros nodos C, por medio del motor de coordinación 208, pueden iniciar un proceso de elección de un nuevo nodo C replicador de bloque.

De hecho, según un ejemplo, una propuesta de replicador de bloque es una vía para que el nodo C que tiene funciones de replicación de bloque confirme su misión como replicador de bloque a otros nodos C a través de BR HB periódicas y como una vía para dirigir una elección de un nuevo replicador de bloque cuando expira la BR HB. Según una realización, una propuesta de replicador de bloque puede comprender un:

- brId: la identificación del nodo C destinado a ser el replicador de bloque

- brAge: el GSN del nodo C propuesto

Cada nodo C puede almacenar el último acuerdo de replicador de bloque que ha recibido y la hora en que dicho acuerdo fue recibido: <*último ARB, último Recibido*>.

Por ejemplo, en el supuesto de que haya tres nodos C nc1, nc2, ncn3, nc1 es el nodo C replicador de bloque actual. El nodo C nc1 propone periódicamente la propuesta de replicador de bloque como una BR HB. Esta propuesta consiste en su propia identidad de nodo nc1 y la nueva edad del replicador de bloque, que es igual al último GSN observado por nc1 en el momento de la propuesta. El motor de coordinación 208 recibe la propuesta de replicador de bloque, genera un acuerdo correspondiente y suministra el acuerdo a todos los nodos C nc1, nc2 y nc3. El nodo nc1, siendo el replicador de bloque actual, conoce el acuerdo y comienza el trabajo de replicación de bloque. Los nodos C nc2 y nc3 no son los actuales replicadores de bloque, ya que solo recuerdan <*último ARB, último Recibido*> y continúan operaciones regulares (no replicaciones). Cuando *último Recibido* excede un umbral configurado, nc2 y/o nc3 puede iniciar la elección del nuevo replicador de bloque proponiéndose a sí mismo, según una realización, como el candidato.

Según un ejemplo, el proceso de elección puede ser iniciado por cualquier nodo C (o por varios de ellos simultáneamente) una vez que el nodo C detecta que la señal de monitorización de replicador de bloque BR HB ha expirado. El nodo C inicial puede, según una realización, iniciar el proceso de elección proponiéndose a sí mismo como un nuevo replicador de bloque. La propuesta puede incluir la identificación del nodo y el último GSN que el nodo C inicial había visto en ese momento. La propuesta puede enviarse al motor de coordinación 208 y cuando el acuerdo correspondiente llega a los otros nodos C, estos actualizan su misión con respecto a la función de replicador de bloque debidamente. Así es como el nodo C que inició el proceso de elección puede convertirse en el nuevo replicador de bloque. Según un ejemplo, en el caso de que varios nodos C inicien la elección simultáneamente, el nodo C que propuso el acuerdo con el mayor GSN se convierte en el replicador de bloque. Por lo tanto, el nodo C que tiene funciones de replicador de bloque puede cambiar varias veces durante el proceso de elección, pero al final habrá solamente un nodo C replicador de bloque y todos los nodos C coincidirán en qué nodo C tiene las funciones de replicador de bloque. Según un ejemplo, se garantiza que un nodo C fallido nunca haga ninguna replicación de bloque o tome decisiones de eliminación incluso si vuelve a activarse en línea después del fallo asumiendo aún que es el replicador de bloque. Esto se debe a que la decisión de replicar o eliminar bloques se toma solamente como resultado del procesamiento de una BR HB. Esto significa que, después de incorporarse al servicio, el nodo C esperará la próxima señal de monitorización de replicador de bloque BR HB para tomar una decisión de replicación, pero el acuerdo de señal de monitorización contendrá información sobre la nueva asignación de replicador de bloque, tras cuya recepción el nodo C activo recientemente conocerá que ya no tiene la función de replicación de bloque.

Para que cada nodo C esté habilitado para generar o permitir la generación de bloques se requiere que cada bloque de datos almacenado en los nodos de datos sea identificable de manera singular, a través de todo el clúster. Los identificadores que generan aleatoriamente bloques de datos de gran caudal (ID) y que luego comprueban si dicho ID de bloque de datos es realmente único es el procedimiento actual para generar ID de bloques en el HDFS. Este enfoque es problemático para nodos C replicados ya que el nuevo ID de bloque debe generarse antes de la propuesta de crear el bloque enviada al motor de coordinación, pero en el momento en que el acuerdo correspondiente llega a los nodos C, el ID podría haber sido asignado ya a otro bloque a pesar de que el ID estaba libre en el momento en que fue generado. La coordinación de dichas colisiones en el momento del acuerdo, aunque posible, añade complejidad, tráfico y atraso innecesarios al proceso y retrasa el reconocimiento eventual al cliente de la generación exitosa del bloque de datos. En cambio, según una realización y como se muestra en la Fig. 6, puede definirse un amplio intervalo, que varía desde un número de ID de bloque mínimo (MINLONG) a un número de ID de bloque máximo (MAXLONG). Este amplio intervalo puede ser tan amplio como se requiera para garantizar que cada número de ID de bloque de datos sea único a través de todo el clúster y, según una realización, supere el tiempo de vida previsto del mismo. Por ejemplo, el intervalo de MINLONG a MAXLONG puede ser, por ejemplo, un número que comprende 1024 *bits* o más. Por lo tanto, para garantizar que cada nodo C genere números de ID de bloque de datos únicos, el intervalo de MINLONG a MAXLONG puede dividirse lógicamente en tres intervalos de ID de bloque de nodo C, mostrados en la Fig. 6 en los intervalos 602, 604 y 606. Por ejemplo, el intervalo de ID de bloque de datos 602 puede abarcar de MINLONG a MINLONG + X *bits*, el intervalo de ID de bloque 604 puede abarcar de MINLONG + X a MINLONG + 2X y el intervalo de ID de bloque 606 puede abarcar de MINLONG + 2X a MAXLONG.

La Fig. 7 es un diagrama de flujo de un procedimiento implementado por ordenador para implementar un sistema de archivo distribuido que comprende una pluralidad de nodos de datos configurados para almacenar bloques de datos de archivos, según un ejemplo. Como se muestra en el bloque B71, el procedimiento puede comprender una etapa

de acoplamiento de al menos tres nodos de nombre (o mayor cantidad de números impares) con una pluralidad de nodos de datos. Cada nodo de nombre puede configurarse, según una realización, para almacenar un estado del espacio de nombre del clúster. Como se muestra en el bloque B72, puede realizarse después la etapa de (el motor de coordinación 208, por ejemplo) recibir propuestas de los nodos de nombre (tal como se muestra en 202, 204, 206 en la Fig. 2) para cambiar el estado del espacio de nombre creando o eliminando archivos y directorios y añadiendo los bloques de datos almacenados en uno o más de la pluralidad de nodos de datos (tal como se muestra en 302, 304 y 306 en la Fig. 3). Dentro de la presente descripción, “cambiar”, donde proceda, comprende añadir nuevos bloques de datos, replicar bloques de datos o eliminar bloques de datos del archivo de un cliente. Como se muestra en B73, el procedimiento implementado por ordenador puede comprender además generar, en respuesta a la recepción de propuestas, un conjunto ordenado de acuerdos que especifica la secuencia de los nodos de nombre para cambiar el estado del espacio de nombre. Según un ejemplo, por lo tanto, los nodos de nombre tardan haciendo cambios (solicitados por clientes, por ejemplo) al estado del espacio de nombre hasta que los nodos de nombre reciban el conjunto ordenado de acuerdos (desde el motor de coordinación 208, por ejemplo).

Según un ejemplo, cuando un nuevo nodo C es puesto en línea (tal como puede ser el caso en el que un nodo C existente ha fallado o de lo contrario se ha desactivado), el nuevo nodo C puede iniciarse en modo seguro, como se ha mencionado anteriormente. El nuevo nodo C en modo seguro puede entonces comenzar a recibir registros e informes de bloques de datos iniciales desde los nodos de datos, identificando los bloques de datos almacenados en cada uno de los nodos de datos a los que se acopla el nuevo nodo C. Según un ejemplo, cuando un nodo C está en modo seguro, no acepta solicitudes de clientes para modificar el estado del espacio de nombre. Esto significa que antes de enviar una propuesta, el nuevo nodo C comprueba si está en modo seguro y expulsa la excepción de modo seguro si el nuevo nodo C determina que actualmente está funcionando en modo seguro. Cuando se recibe un número suficiente de informes de bloque, según un ejemplo, el nuevo nodo C puede abandonar el modo seguro y comenzar a aceptar solicitudes de modificación de datos de los clientes. Al inicio, según un ejemplo, los nodos C entran automáticamente en el modo seguro y luego automática y asincrónicamente abandonan también el modo seguro una vez que han recibido un número suficiente de informes de réplicas de bloques. La salida del modo seguro automático, según un ejemplo, no se coordina a través del motor de coordinación 208, porque los nodos C (tales como los nodos C 202, 204 y 206 en la Fig. 2) pueden procesar informes de bloques en diferentes intervalos y, por lo tanto, pueden alcanzar el umbral al que deben salir del modo seguro en momentos diferentes. En cambio, cuando un administrador de clúster emite un comando para entrar en modo seguro, todos los nodos C deben obedecer. Por esta razón, los comandos de modo seguro emitidos por administrador pueden coordinarse, según un ejemplo, a través del motor de coordinación 208.

Como se ha mencionado anteriormente, los nodos C pueden fallar o deshabilitarse intencionalmente para su mantenimiento. Según una realización, los nodos C replicados restantes seguirán funcionando mientras formen un *quorum* suficiente para que el motor de coordinación 208 genere acuerdos. Si se pierde el *quorum*, según una realización, el clúster se bloquearía y dejaría de procesar solicitudes de cambios de espacio de nombre hasta que se restaure el *quorum*.

Cuando un nodo C que ha fallado anteriormente o un nodo C que fue deshabilitado deliberadamente vuelve a estar en línea, recuperará automáticamente los otros nodos C en su estado. Según una realización, el motor de coordinación 208 puede suministrar al nodo C que se volvió a poner en línea todos los acuerdos que pudo haber ignorado mientras estaba deshabilitado. Durante este periodo de tiempo, el nodo C que se vuelve a poner en línea no tiene su servidor de RPC iniciado. Por lo tanto, los clientes y los nodos de datos no son capaces de conectarse al mismo (ya que la RPC es el modo a través del cual deben comunicarse), lo que evita que el nodo C que se vuelve a habilitar suministre datos potencialmente obsoletos a los clientes solicitantes. Este proceso tiene lugar antes de que los nodos de datos se conecten con el nodo C que se ha puesto en línea nuevamente. Los registros de nodo de datos y los informes de bloque iniciales deben retrasarse ya que los informes pueden contener bloques que el nodo C no conoce todavía y que podrían haber sido descartados de sus informes.

Si el nodo C estuviese deshabilitado durante largo tiempo y hubiese ignorado un número considerable de acuerdos (que puede ser un umbral configurable), puede resultar inviable e irrealizable esperar hasta que el nodo C reciba los acuerdos que ignoró mientras estaba deshabilitado y reproducir el historial completo de acuerdos ignorados. En este caso y según una realización, puede ser más eficiente hacer que el nodo C descargue un punto de control desde uno de los nodos C activos, lo cargue como el estado de espacio de nombre inicial y seguidamente reciba acuerdos desde el motor de coordinación 208 empezando a partir del punto de control y reproduzca luego el historial de los acuerdos provistos desde el momento en que se realizó el punto de control. Para hacerlo de este modo, el nodo C que se volvió a poner en línea puede escoger uno de los nodos activos (llamado el “ayudante”) como una fuente para recuperar el punto de control y envía una solicitud de RPC (por ejemplo, *iniciarPuntodecontrol()*) al nodo C ayudante escogido. El nodo C ayudante luego emite una propuesta de inicio de punto de control al motor de coordinación 208, para garantizar que todos los nodos C se sincronicen con sus puntos de control locales en el mismo GSN. Cuando el acuerdo de inicio de punto de control llega, el nodo C ayudante recordará el GSN de dicho acuerdo como un punto de control específicamente identificado que es válido para un GSN específico (por ejemplo, GSN de punto de control). Este GSN de punto de control determina luego el acuerdo según el cual el nodo C emergente iniciará el proceso de aprendizaje una vez que use el punto de control.

El consumo del punto de control por el nodo C que ha sido puesto nuevamente en línea puede realizarse mediante

la carga de la imagen y los archivos de diario, como lo estándar para HDFS. Después de la recuperación, el nodo C puede entonces iniciar la recepción de informes de bloques desde los nodos de datos. Una vez que el modo seguro está desactivado, el nodo C que se ha activado en línea nuevamente puede unirse totalmente al clúster y reanudar sus funciones normales.

Según un ejemplo, el inicio de un nuevo nodo C o un reinicio de un nodo C existente puede comprender las siguientes etapas principales.

1. El nodo C puesto en línea nuevamente se inicia y se une al clúster como un proponente, pero con las capacidades de aprendiz desactivadas hasta la etapa 3.

a) Analiza su estado en el historial global en relación con otros nodos.

2. Si su estado está sustancialmente por detrás de otros nodos, determinado por un umbral configurable, entonces descargará un punto de control más reciente de uno seleccionado de los nodos ayudantes activos. El nodo ayudante seleccionado también proporciona el GSN de punto de control, que corresponde al estado en el historial a partir de la creación del punto de control.

3. Cuando se descarga el punto de control (si fue necesario), el nodo C puesto nuevamente en línea envía su primera propuesta al motor de coordinación 208, llamada propuesta de recuperación de acuerdos (PRA) y asume el papel de aprendiz.

a) El nodo C puesto nuevamente en línea puede empezar a conocer los acuerdos que ignoró mientras estaba deshabilitado, empezando por el GSN de punto de control + 1.

4. Cuando el nodo C puesto nuevamente en línea llega a su propio primer acuerdo PRA, el proceso de actualización se considera completado. El nodo C puesto nuevamente en línea puede asumir ahora el papel de aceptador y convertirse en un participante completamente funcional del clúster y recibir acuerdos adicionales desde, y enviar propuestas al, motor de coordinación 208.

5. Para ello, el nodo C puede inicializar su servidor de RPC y ponerse a sí mismo a la disponibilidad de nodos de datos para registros e informes de bloques. Después del procesamiento de los informes y abandonar el modo seguro, el nodo C puede comenzar a aceptar las solicitudes de cliente en igualdad de condiciones con respecto a otros nodos C del clúster.

Como se ha mencionado anteriormente, cada nodo C, según un ejemplo, puede almacenar una imagen del espacio de nombre y actualizarla en un almacenamiento persistente local (no volátil) que se acopla al nodo C. Debe tenerse en cuenta que el almacenamiento local (si existiese) puede configurarse de modo que no se comparta entre nodos C. Según un ejemplo, cada nodo C puede mantener en su almacenamiento persistente local, su propio archivo de imagen local que contiene un último punto de control de imagen de espacio de nombre y un archivo de edición local, cuyo archivo de edición constituye un diario de transacciones aplicado al espacio de nombre desde el último punto de control. Según un ejemplo, la desactivación de un clúster puede inhabilitar los nodos C en diferentes momentos de la evolución del espacio de nombre. Esto significa que algunos nodos C pueden haber aplicado ya toda la transacción especificada por los acuerdos recibidos desde el motor de coordinación 208, aunque algunos nodos C rezagados pueden iniciarse en un estado anterior al estado actual. Sin embargo, el motor de coordinación 208 puede configurarse para impulsar el nodo C rezagado hacia el estado actual suministrándole los eventos ignorados desde la secuencia global.

Debe tenerse en cuenta que esto no difiere de la operación de clúster nominal cuando algunos nodos C pueden caer detrás de otros al actualizar el estado del espacio de nombre a lo largo del procesamiento de acuerdos recibidos desde el motor de coordinación 208. Dichos nodos C rezagados pueden aceptar todavía solicitudes de modificación de espacio de nombre de los clientes y hacer propuestas al motor de coordinación 208. Las propuestas resultantes se ordenarán, se colocarán en la secuencia global después de los eventos que el nodo C todavía tiene que procesar y se aplicarán para actualizar el estado del espacio de nombre en su debido orden. De esta manera, un nodo C rezagado puede “recuperar la velocidad” (o sea, alcanzar el GSN más actual), antes de procesar nuevas solicitudes, manteniendo así la consistencia en el estado del espacio de nombre a través de los nodos C del clúster. Según una realización, las discrepancias en el estado persistente de nodos C durante el inicio pueden evitarse realizando un procedimiento de apagado “limpio”.

Según un ejemplo, puede proporcionarse un procedimiento de apagado limpio para forzar a todos los nodos C a un estado común antes del apagado del clúster. Como resultado de la realización de un apagado limpio, todas las imágenes locales del espacio de nombre almacenadas en la memoria local persistente acoplada a cada uno de los nodos C serán idénticas y las actualizaciones de las mismas pueden representarse mediante una secuencia vacía de transacciones. Según un ejemplo, para apagar limpiamente y forzar todas las imágenes locales del espacio de nombre a que sean idénticas, puede ordenarse a cada nodo C que entre en el modo seguro de operación, durante cuyo tiempo el nodo C deja de procesar solicitudes de cliente para modificar el espacio de nombre, mientras que los acuerdos restantes enviados al mismo por el motor de coordinación 208 siguen siendo procesados. Posteriormente, los procesos de nodo C pueden aniquilarse. Después de un apagado limpio, cualquier proceso de inicio posterior

procederá con más rapidez que si no se hubiesen apagado los nodos C limpiamente, ya que ningún nodo C necesita aplicar ediciones y actualizaciones ignoradas desde el motor de coordinación 208 (ya que todos se colocaron en un estado idéntico previo al apagado).

Aunque se han descrito algunas realizaciones de la descripción, dichas realizaciones se han presentado solamente a modo de ejemplo. Por ejemplo, un ejemplo comprende un medio legible por máquina no transitorio, tangible, que tiene datos almacenados en el mismo que representan secuencias de instrucciones que, cuando son ejecutadas por dispositivos informáticos, hacen que los dispositivos informáticos implementen un sistema de archivo distribuido tal como se describe y muestra en esta memoria. Por ejemplo, las secuencias de instrucciones pueden descargarse y luego almacenarse en un dispositivo de memoria (tal como se muestra en 702 en la Fig. 7, por ejemplo). Las instrucciones pueden comprender, según un ejemplo, instrucciones para acoplar al menos dos nodos de nombre a una pluralidad de nodos de datos, estando configurados cada uno de los nodos de nombre para almacenar un estado de un espacio de nombre del clúster y estando configurado cada uno para responder a una solicitud de un cliente mientras al menos otro de los nodos de nombre responde a otra solicitud de otro cliente; recibir propuestas de los nodos de nombre para cambiar el estado del espacio de nombre replicando, eliminando y/ o añadiendo bloques de datos en/ desde/ a uno o más de la pluralidad de nodos de datos y generando, como respuesta a la recepción de propuestas, un conjunto ordenado de acuerdos que especifica una orden en la que los nodos de nombre deben cambiar el estado del espacio de nombre, de modo que los nodos de nombre retrasan la realización de cambios en el estado del espacio de nombre hasta que los nodos de nombre reciben el conjunto ordenado de acuerdos. El alcance de la presente descripción pretende ser definida solamente con referencia a las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un clúster de nodos que comprende dispositivos informáticos configurados para implementar un sistema de archivo distribuido, que comprende:

una pluralidad de nodos de datos (302, 304, 306, 402), configurado cada uno para almacenar bloques de datos de archivos de clientes (350);

al menos dos nodos de nombre (202, 204), cada uno acoplado a la pluralidad de nodos de datos en una red informática, estando configurado cada nodo de nombre para almacenar una imagen de un estado de un espacio de nombre del clúster y estando configurado cada uno para responder a una solicitud para cambiar su respectiva imagen almacenada del estado del espacio de nombre de un cliente mientras que al menos otro de los nodos de nombre responde a otra solicitud para cambiar su respectiva imagen almacenada del estado del espacio de nombre del clúster de otro cliente;

un motor de coordinación (208) acoplado a cada uno de los al menos dos nodos de nombre, estando configurado el motor de coordinación para recibir propuestas de al menos dos nodos de nombre para cambiar sus respectivas imágenes del estado del espacio de nombre y para generar, como respuesta, un conjunto ordenado de acuerdos que especifica un orden en el que al menos dos nodos de nombre deben cambiar sus respectivas imágenes almacenadas del estado del espacio de nombre, donde cada uno de los al menos dos nodos de nombre está configurado para retrasar la realización de cambios en su respectiva imagen almacenada del estado del espacio de nombre hasta recibir la recepción del conjunto ordenado de acuerdos.

2. El clúster de nodos de la reivindicación 1, donde cada nodo de nombre de los al menos dos nodos de nombre (202, 204) está configurado para generar una propuesta en respuesta a un cliente que emite una solicitud para al menos replicar o eliminar los bloques de datos en los nodos de datos.

3. El clúster de la reivindicación 1 o 2, donde el motor de coordinación está configurado además para asignar un número de secuencia global único (GSN) a cada acuerdo, especificando el GSN un orden en el que al menos dos nodos de nombre (202, 204) deben aplicar cambios a su respectiva imagen almacenada del estado del espacio de nombre.

4. El clúster de la reivindicación 3, donde los al menos dos nodos de nombre (202, 204) están configurados además para aplicar los cambios a su respectiva imagen almacenada del estado del espacio de nombre en el orden especificado por el GSN de los acuerdos recibidos.

5. El clúster de cualquiera de las reivindicaciones 1 a 3, que comprende además el almacenamiento local (210, 212) acoplado a cada uno de los al menos dos nodos de nombre (202, 204), estando configurado el almacenamiento local para almacenar la imagen del espacio de nombre y las entradas que especifican las actualizaciones a la imagen almacenada.

6. El clúster de la reivindicación 1, donde solo un nodo de nombre de los al menos dos nodos de nombre (202, 204) en el clúster está configurado para permitir la replicación y eliminación de bloques de datos almacenados en la pluralidad de nodos de datos.

7. El clúster de cualquiera de las reivindicaciones 1 a 6, donde todos los nodos de nombre (202, 204) en el clúster están configurados para soportar activamente las solicitudes de los clientes simultáneamente.

8. Un procedimiento implementado por ordenador para implementar un sistema de archivo distribuido que comprende una pluralidad de nodos de datos (302, 304, 306, 402) configurados para almacenar bloques de datos de archivos de clientes (350), comprendiendo el procedimiento:

acoplar al menos dos nodos de nombre (202, 204) a una pluralidad de nodos de datos en una red informática, estando configurados cada uno de los al menos dos nodos de nombre para almacenar una imagen de un estado de un espacio de nombre del clúster y estando configurado cada uno para responder a una solicitud para cambiar su respectiva imagen almacenada del estado del espacio de nombre de un cliente, mientras que al menos otro de los al menos dos nodos de nombre responde a otra solicitud para cambiar su respetiva imagen almacenada del estado del espacio de nombre del clúster de otro cliente;

recibir propuestas de los al menos dos nodos de nombre (202, 204) para cambiar sus respectivas imágenes almacenadas del estado del espacio de nombre; y

generar, en respuesta a la recepción de las propuestas, un conjunto ordenado de acuerdos que especifica un orden en el que los al menos dos nodos de nombre deben cambiar sus respectivas imágenes almacenadas del estado del espacio de nombre, de modo que los al menos dos nodos de nombre retrasen la realización de cambios en sus respectivas imágenes almacenadas del estado del espacio de nombre hasta la recepción del conjunto ordenado de acuerdos.

9. El procedimiento implementado por ordenador de la reivindicación 8, que comprende además los al menos dos

nodos de nombre (202, 204) que generan las propuestas que responden a los clientes que envían solicitudes para al menos replicar o eliminar bloques de datos en los nodos de datos (302, 304, 306, 402).

10. El procedimiento implementado por ordenador de la reivindicación 8 o 9, que comprende además asignar un número de secuencia global único (GSN) a cada acuerdo del conjunto ordenado de acuerdos, especificando el GSN un orden en el que los al menos dos nodos de nombre (202, 204) deben aplicar los cambios a su respectiva imagen almacenada del estado del espacio de nombre.

11. El procedimiento implementado por ordenador de la reivindicación 10, que comprende además los al menos dos nodos de nombre (202, 204) que aplican los cambios a su respectiva imagen almacenada del estado del espacio de nombre en el orden especificado por el GSN de los acuerdos recibidos.

12. El procedimiento implementado por ordenador de cualquiera de las reivindicaciones 8 a 11, que comprende además configurar solo uno de los al menos dos nodos de nombre (202, 204) en el clúster para permitir que los bloques de datos almacenados en la pluralidad de nodos de datos se repliquen o eliminen.

13. El procedimiento implementado por ordenador de cualquiera de las reivindicaciones 8 a 12, que comprende además configurar cada uno de los al menos dos nodos de nombre (202, 204) para permitir que se generen y almacenen bloques de datos en la pluralidad de nodos de datos.

14. El procedimiento implementado por ordenador de cualquiera de las reivindicaciones 8 a 13, que comprende además configurar cada uno de los nodos de nombre (202, 204) para permitir que los bloques de datos almacenados en uno de la pluralidad de nodos de datos se repliquen a uno predeterminado o a otros de la pluralidad de nodos de datos.

15. El procedimiento implementado por ordenador de cualquiera de las reivindicaciones 8 a 14, que comprende además habilitar todos los nodos de nombre (202, 204) en el clúster para soportar activamente las solicitudes de los clientes simultáneamente.

16. El procedimiento implementado por ordenador de cualquiera de las reivindicaciones 8 a 15, que comprende además permitir que un nuevo nodo de nombre (206) se incorpore al clúster, donde el nuevo nodo de nombre está configurado para actualizar el estado de su imagen almacenada del espacio de nombre mediante al menos:

la carga de un punto de control que comprende una imagen anterior del espacio de nombre, y

la aceptación de acuerdos para su imagen almacenada del espacio de nombre del motor de coordinación.

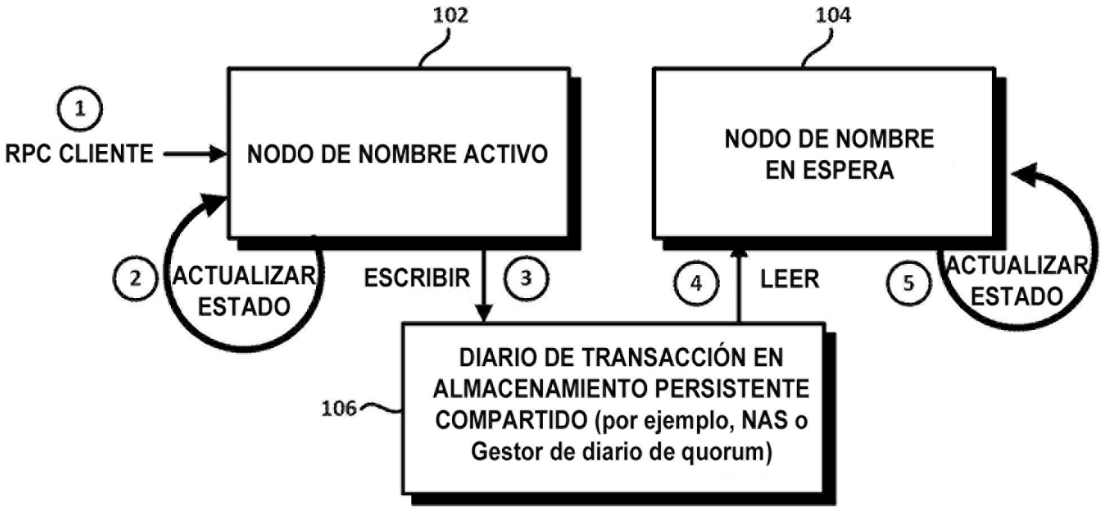

FIG. 1 (Técnica anterior)

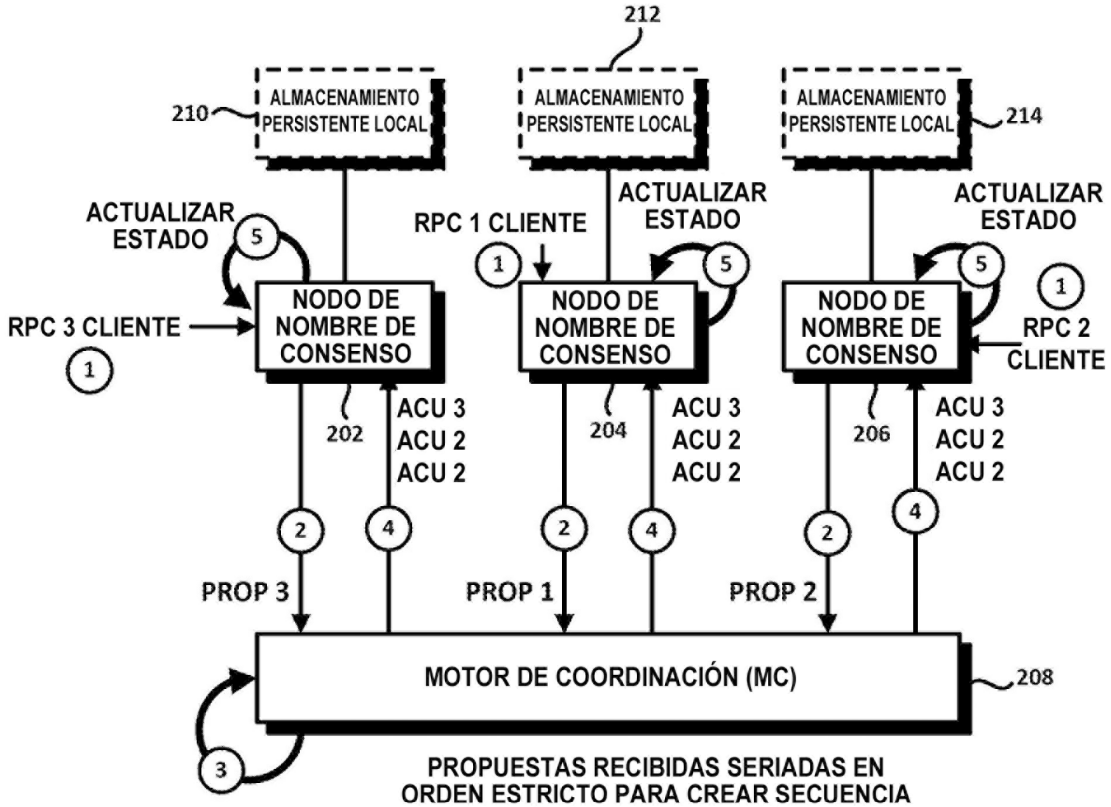
212
ALMACENAMIENTO PERSISTENTE LOCAL
210
ALMACENAMIENTO PERSISTENTE LOCAL
ALMACENAMIENTO PERSISTENTE LOCAL
214
ACTUALIZAR ESTADO
RPC 1 CLIENTE
ACTUALIZAR ESTADO
ACTUALIZAR ESTADO
RPC 3 CLIENTE
NODO DE NOMBRE DE CONSENSO
NODO DE NOMBRE DE CONSENSO
NODO DE NOMBRE DE CONSENSO
RPC 2 CLIENTE
202
204
206
ACU 3
ACU 2
ACU 2
PROP 3
PROP 1
PROP 2
MOTOR DE COORDINACIÓN (MC)
208
PROPUESTAS RECIBIDAS SERIADAS EN ORDEN ESTRICTO PARA CREAR SECUENCIA GLOBAL DE EVENTOS

FIG. 2

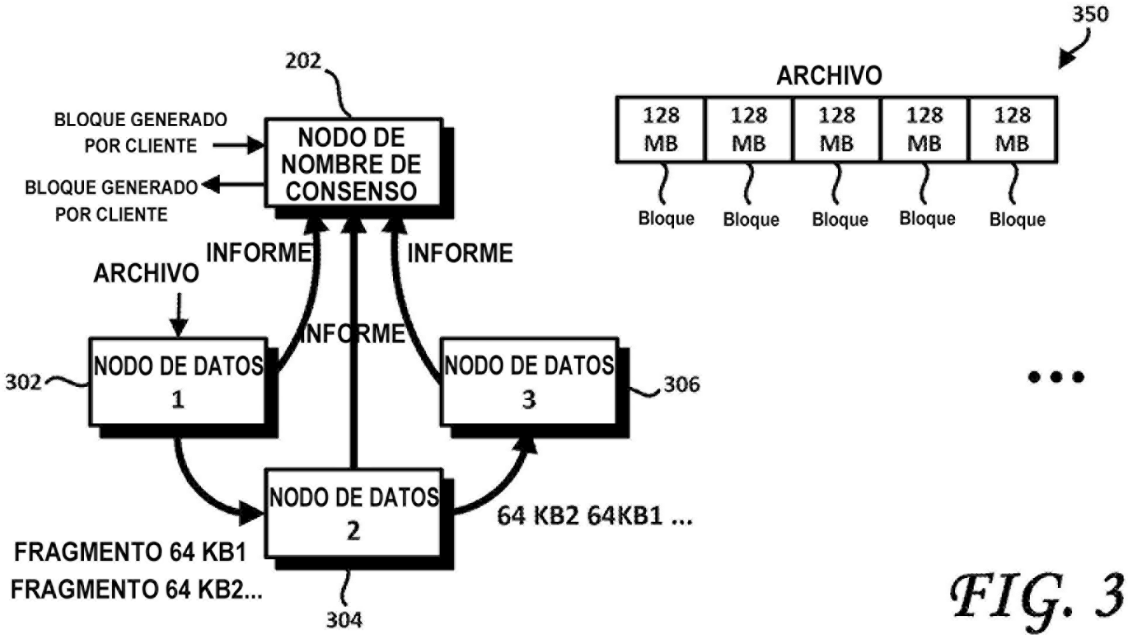
350
ARCHIVO
128 MB
Bloque
202
BLOQUE GENERADO POR CLIENTE
BLOQUE GENERADO POR CLIENTE
NODO DE NOMBRE DE CONSENSO
INFORME
INFORME
ARCHIVO
INFORME
NODO DE DATOS 1
NODO DE DATOS 3
NODO DE DATOS 2
302
306
304
64 KB2 64KB1 ...
FRAGMENTO 64 KB1
FRAGMENTO 64 KB2...

FIG. 3

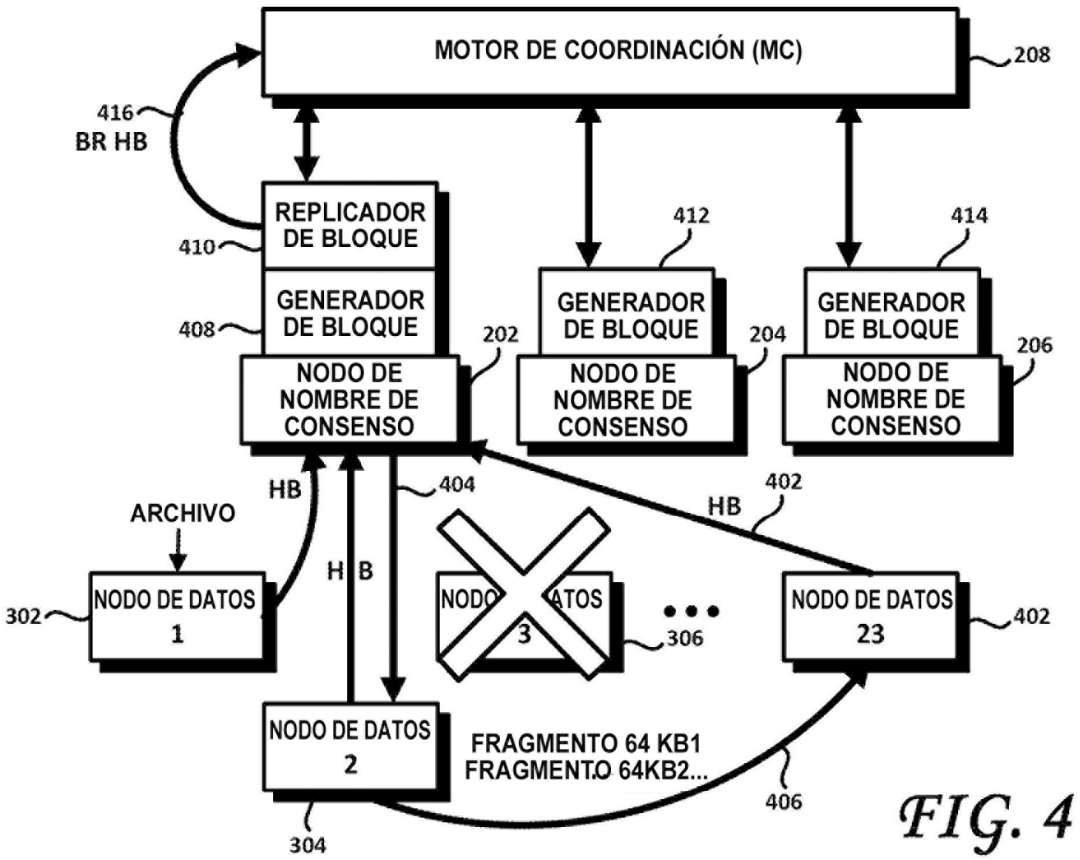
MOTOR DE COORDINACIÓN (MC)
208
416
BR HB
REPLICADOR DE BLOQUE
410
GENERADOR DE BLOQUE
408
NODO DE NOMBRE DE CONSENSO
202
412
GENERADOR DE BLOQUE
NODO DE NOMBRE DE CONSENSO
204
414
GENERADOR DE BLOQUE
NODO DE NOMBRE DE CONSENSO
206
HB
404
HB
402
ARCHIVO
NODO DE DATOS 1
302
H B
NODO DE DATOS 3
306
NODO DE DATOS 23
402
NODO DE DATOS 2
304
FRAGMENTO 64 KB1
FRAGMENTO 64KB2...
406

FIG. 4

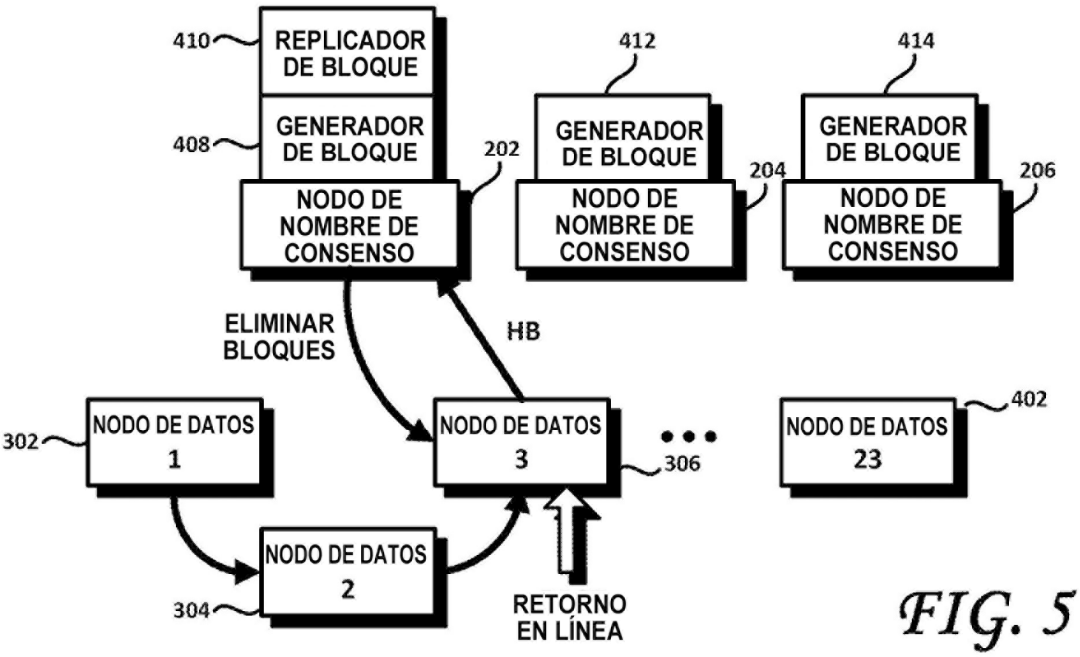
410
REPLICADOR DE BLOQUE
408
GENERADOR DE BLOQUE
NODO DE NOMBRE DE CONSENSO
202
412
GENERADOR DE BLOQUE
NODO DE NOMBRE DE CONSENSO
204
414
GENERADOR DE BLOQUE
NODO DE NOMBRE DE CONSENSO
206
ELIMINAR BLOQUES
HB
302
NODO DE DATOS 1
NODO DE DATOS 3
306
NODO DE DATOS 23
402
NODO DE DATOS 2
304
RETORNO EN LÍNEA

FIG. 5

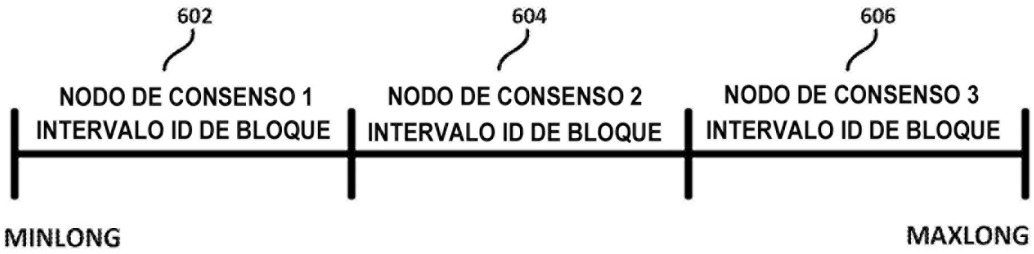
602
604
606
NODO DE CONSENSO 1
INTERVALO ID DE BLOQUE
NODO DE CONSENSO 2
INTERVALO ID DE BLOQUE
NODO DE CONSENSO 3
INTERVALO ID DE BLOQUE
MINLONG
MAXLONG

FIG. 6

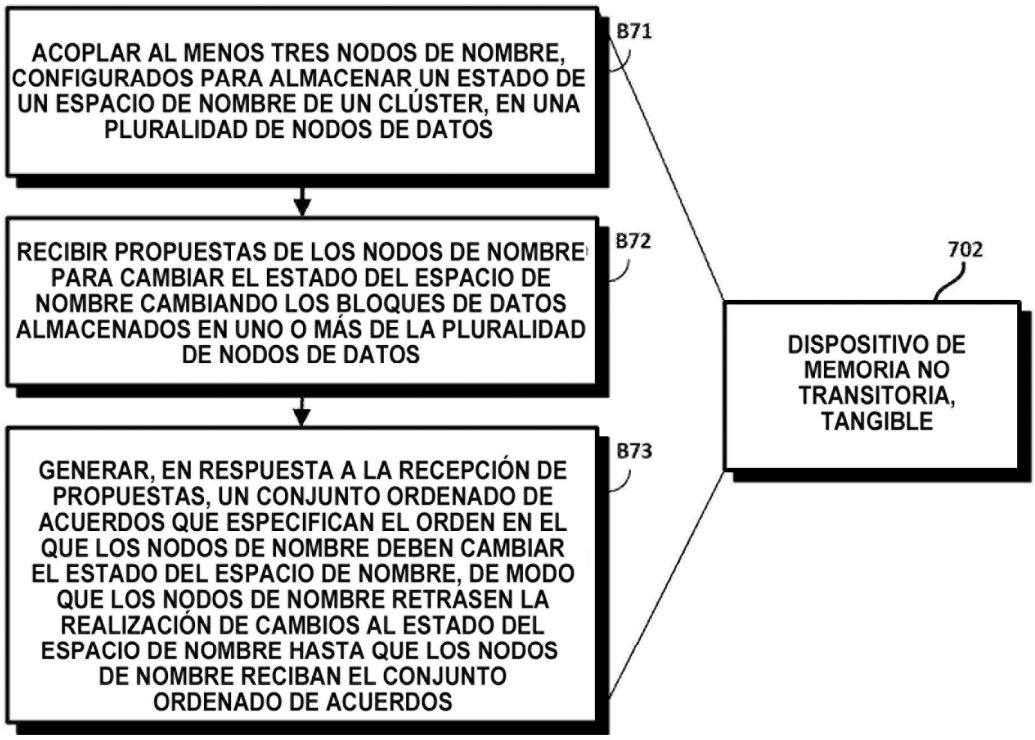
ACOPLAR AL MENOS TRES NODOS DE NOMBRE, CONFIGURADOS PARA ALMACENAR UN ESTADO DE UN ESPACIO DE NOMBRE DE UN CLÚSTER, EN UNA PLURALIDAD DE NODOS DE DATOS
B71
RECIBIR PROPUESTAS DE LOS NODOS DE NOMBRE PARA CAMBIAR EL ESTADO DEL ESPACIO DE NOMBRE CAMBIANDO LOS BLOQUES DE DATOS ALMACENADOS EN UNO O MÁS DE LA PLURALIDAD DE NODOS DE DATOS
B72
GENERAR, EN RESPUESTA A LA RECEPCIÓN DE PROPUESTAS, UN CONJUNTO ORDENADO DE ACUERDOS QUE ESPECIFICAN EL ORDEN EN EL QUE LOS NODOS DE NOMBRE DEBEN CAMBIAR EL ESTADO DEL ESPACIO DE NOMBRE, DE MODO QUE LOS NODOS DE NOMBRE RETRASEN LA REALIZACIÓN DE CAMBIOS AL ESTADO DEL ESPACIO DE NOMBRE HASTA QUE LOS NODOS DE NOMBRE RECIBAN EL CONJUNTO ORDENADO DE ACUERDOS
B73
702
DISPOSITIVO DE MEMORIA NO TRANSITORIA, TANGIBLE

FIG. 7