(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 880 448**

(51) Int. CI.:

G06F 11/20 (2006.01)
G06F 11/07 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **06.03.2018 PCT/CN2018/078169**

(87) Fecha y número de publicación internacional: **13.09.2018 WO18161901**

(96) Fecha de presentación y número de la solicitud europea: **06.03.2018 E 18764913 (2)**

(97) Fecha y número de publicación de la concesión europea: **05.05.2021 EP 3525102**

(54) Título: **Procedimiento y dispositivo de consenso**

(30) Prioridad:

10.03.2017 CN 201710142252

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
24.11.2021

(73) Titular/es:

ADVANCED NEW TECHNOLOGIES CO., LTD. (100.0%)
Cayman Corporate Centre, 27 Hospital Road George Town, Grand Cayman KY1-9008, KY

(72) Inventor/es:

TANG, QIANG

(74) Agente/Representante:

LEHMANN NOVO, María Isabel

DESCRIPCIÓN

Procedimiento y dispositivo de consenso

CAMPO TÉCNICO

La presente patente se refiere al campo de las tecnologías informáticas y, en particular, a un procedimiento y un aparato de consenso.

ANTECEDENTES

Actualmente, la tecnología de cadena de bloques se utiliza ampliamente, y un modo descentralizado en la tecnología de cadena de bloques garantiza que los datos no se manipulen fácilmente, mejorándose así la seguridad.

En la práctica, una red de cadena de bloques que incluye múltiples nodos (los nodos pueden considerarse dispositivos de la red de cadena de bloques que participan en servicios de procesamiento) puede proporcionar servicios correspondientes para dispositivos cliente. Los nodos de la red de cadena de bloques realizan un procesamiento de solicitudes de servicio de los dispositivos cliente y retroalimentan los resultados de procesamiento a los dispositivos cliente. En este proceso, los resultados de procesamiento generados por nodos que se ejecutan de forma independiente pueden ser incoherentes. Para garantizar que el dispositivo cliente pueda recibir un resultado de procesamiento correcto, se implementa un consenso entre los nodos mediante el uso de un algoritmo de Tolerancia Práctica a Fallas Bizantinas (PBFT) (es decir, un resultado de procesamiento correcto puede ser reconocido o aceptado conjuntamente por los nodos).

En un proceso de uso del algoritmo PBFT, generalmente se realiza un consenso en una vista. En una vista, un nodo de una red de cadena de bloques se utiliza como nodo primario (primario), y otros nodos se utilizan como nodos de respaldo (respaldo). En este caso, el nodo primario recibe una solicitud de servicio del dispositivo cliente y difunde la solicitud de servicio a todos los nodos de respaldo, y el nodo primario inicia un consenso. Los nodos que alcanzan el consenso realizan el procesamiento de la solicitud de servicio y retroalimentan un resultado de procesamiento al dispositivo cliente.

En la tecnología existente, un nodo de respaldo inicia una conmutación de vista, y la conmutación de vista iniciada por el nodo de respaldo generalmente necesita ser reconocida o aceptada por otros nodos en la vista. El nodo de respaldo inicia una solicitud de conmutación de vista a otros nodos (incluido el nodo primario) en la vista, es decir, inicia un consenso sobre la solicitud de conmutación de vista a otros nodos (este consenso sigue utilizando PBFT. A diferencia del proceso del consenso sobre la solicitud de servicio, en un proceso de consenso basado en la solicitud de conmutación de vista, cada nodo suspende el consenso sobre la solicitud de servicio. Por lo tanto, el consenso sobre la solicitud de conmutación de vista es esencialmente un proceso de consenso adicional). Después de que una cantidad predeterminada de nodos alcance un consenso, se determina que un nodo de respaldo pase a ser un nuevo nodo primario. El nuevo nodo primario difunde un nuevo mensaje de vista para completar la conmutación de vista.

Sin embargo, en el mecanismo anterior, es necesario realizar un proceso de consenso adicional para la conmutación de vista iniciada por el nodo de respaldo, y el proceso de consenso adicional aumenta una cantidad de cálculo del sistema. Además, en el proceso de consenso de conmutación de vista, se puede alcanzar un consenso después de ser confirmado por una cantidad predeterminada de nodos. Finalmente, un nuevo nodo primario difunde un nuevo mensaje de vista, y todo el proceso consume un período de tiempo. Evidentemente, una forma de conmutación de vista existente no solo aumenta una cantidad de cálculo del sistema, sino que también aumenta el tiempo consumido para procesar una solicitud de servicio. En consecuencia, se produce una eficiencia de procesamiento relativamente baja.

El documento US 2015/186229 A1 describe una reselección tras la conmutación por error de un líder en un protocolo de consenso tolerante a fallas en un sistema basado en RSM.

El documento WO 2017/186317 A1 constituye la técnica anterior en virtud del Artículo 54(3) EPC y describe la replicación de tolerancia a fallas bizantinas de datos en una pluralidad de n servidores por un cliente, donde los servidores incluyen un nodo primario y n-1 nodos de réplica.

RESUMEN

Implementaciones de la presente patente proporcionan un procedimiento y un aparato de consenso para mitigar el problema de que una forma de conmutación de vista actual aumenta una cantidad de cálculo de una red de cadena de bloques y aumenta el consumo de tiempo de procesamiento.

La presente invención está definida por las reivindicaciones independientes adjuntas.

Las implementaciones de la presente patente proporcionan un procedimiento y un aparato de consenso. En cualquier vista, el nodo primario de cadena de bloques supervisa activamente la activación de la condición de conmutación de vista. Si se activa la condición de conmutación de vista, el nodo primario de cadena de bloques debe realizar la conmutación de vista. Además, el nodo primario de cadena de bloques selecciona un nodo sucesor de entre otros nodos de cadena de bloques como nodo primario de cadena de bloques en la siguiente vista. Por consiguiente, el nodo primario de cadena de bloques realiza la conmutación de vista. En la vista conmutada, el nodo sucesor se utiliza como nuevo nodo primario de cadena de bloques para procesar un servicio. Además, la conmutación de vista se sigue realizando en función del proceso anterior. Evidentemente, el nodo primario de cadena de bloques inicia la conmutación de vista anterior. De esta manera se evita que el nodo de respaldo de cadena de bloques inicie el consenso de conmutación de vista. En otras palabras, puede evitarse un consenso adicional. Por tanto, se puede reducir una cantidad de cálculo adicional y el consumo de tiempo de procesamiento en una red de cadena de bloques.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Los dibujos adjuntos descritos en el presente documento tienen por objeto proporcionar un mejor entendimiento de la presente patente, y constituyen una parte de la presente patente. Las implementaciones ilustrativas de la presente patente y las descripciones de la misma tienen por objeto describir la presente patente, y no constituyen limitaciones en la presente patente. En los dibujos adjuntos:

La Fig. 1a ilustra una arquitectura en la que se basa un proceso de consenso, de acuerdo con una implementación de la presente patente;

la FIG. 1b ilustra un proceso de consenso, de acuerdo con una implementación de la presente patente;

la Fig. 2 es un diagrama esquemático que ilustra un proceso de consenso que se basa en un protocolo de tres fases en cualquier vista, de acuerdo con una implementación de la presente patente;

la Fig. 3 es un diagrama esquemático que ilustra un proceso de ejecución de una instancia de aplicación de conmutación de vista, de acuerdo con una implementación de la presente patente; y

la FIG. 4 es un diagrama estructural esquemático que ilustra un aparato de consenso, de acuerdo con una implementación de la presente patente.

DESCRIPCIÓN DE IMPLEMENTACIONES

Para aclarar los objetivos, las soluciones técnicas y las ventajas de la presente patente, a continuación, se describen de forma clara y completa las soluciones técnicas de la presente patente con referencia a implementaciones específicas y a los dibujos adjuntos de la presente patente. Evidentemente, las implementaciones descritas son sólo algunas y no todas las implementaciones de la presente patente. Todas las demás implementaciones obtenidas por un experto en la técnica en base a las implementaciones de la presente patente sin realizar investigaciones adicionales estarán dentro del alcance de protección de la presente patente.

Como se describió anteriormente, en un proceso en el que los nodos de una red de cadena de bloques realizan un consenso mediante el algoritmo PBFT, una vez que un nodo primario de cadena de bloques se vuelve defectuoso, un nodo de respaldo de cadena de bloques inicia una conmutación de vista. Se necesita un consenso adicional para la conmutación de vista iniciada por el nodo de respaldo de cadena de bloques. Es decir, la conmutación de vista solo se puede completar después de ser reconocida o aceptada por otros nodos de cadena de bloques. Evidentemente, un proceso de consenso adicional aumenta una cantidad de cálculo de una red de cadena de bloques y también aumenta el consumo de tiempo de procesamiento.

Por consiguiente, una implementación de la presente patente proporciona un procedimiento de consenso. En lo que respecta a un nodo primario de cadena de bloques en cualquier vista, después de que finaliza un consenso, el nodo primario de cadena de bloques inicia la conmutación de vista, para cambiar un nodo primario de cadena de bloques, y no se necesita ningún proceso de consenso adicional. Para facilitar la descripción, un nodo primario de cadena de bloques se denomina simplemente nodo primario, y un nodo de respaldo de cadena de bloques se denomina simplemente nodo de respaldo. Además, los "nodos" descritos a continuación deben entenderse como nodos de una red de cadena de bloques que participan en un consenso.

Cabe destacar que, en una implementación de la presente patente, una arquitectura utilizada en el procedimiento de consenso se muestra en la FIG. 1a. En la FIG. 1a se puede observar que una red de cadena de bloques incluye múltiples nodos, y múltiples dispositivos cliente pueden realizar una interacción de servicio con la red de cadena de bloques. Un tipo de aplicación de la red de cadena de bloques puede ser una red de cadena de bloques de consorcio o una red de cadena de bloques privada, y la red de cadena de bloques puede proporcionar un servicio para un usuario. El nodo incluye, pero no se limita a, un servidor, un ordenador, un dispositivo móvil y otros dispositivos que tengan una función de procesamiento informático. El dispositivo cliente puede estar configurado para ejecutar un navegador,

una aplicación, etc., y el dispositivo cliente puede ser un dispositivo de usuario final, un servidor o una base de datos. Las implementaciones no están limitadas aquí.

Basándose en la arquitectura de relación mostrada en la Fig. 1a, un proceso de consenso proporcionado en una implementación de la presente patente se muestra en la FIG. 1b. El proceso incluye las siguientes etapas.

S101: Un nodo primario supervisa la activación de una condición de conmutación de vista.

En esta implementación de la presente patente, la condición de conmutación de vista puede considerarse una condición que debe satisfacerse para realizar la conmutación de vista. Por ejemplo, el nodo primario no difunde una solicitud de servicio ni completa un consenso dentro de un período de tiempo.

En una posible manera en esta implementación de la presente patente, se puede establecer un temporizador en el nodo primario para supervisar la activación de la condición de conmutación de vista. Por ejemplo, se puede usar un temporizador para temporizar un comportamiento de difusión una solicitud de servicio por el nodo primario, para detectar si el comportamiento de difundir una solicitud de servicio por el nodo primario expira. El temporizador puede considerarse una función de temporización o servicio que se ejecuta en el nodo primario y, ciertamente, no constituye limitaciones a la presente patente.

S102: El nodo primario selecciona un nodo sucesor cuando se supervisa la activación de la condición de conmutación de vista.

Si se activa la condición de conmutación de vista, el nodo primario debe realizar la conmutación de vista. Cabe destacar que, en cualquier vista, solo hay un nodo primario, y otros nodos son nodos de respaldo. Por lo tanto, la conmutación de vista indica la conmutación del nodo primario. De este modo, en la presente etapa, el nodo primario selecciona un nodo sucesor como siguiente nodo primario (el nodo sucesor en esta implementación de la presente patente no es el mismo nodo que el nodo primario en la vista actual, es decir, el nodo primario en la vista actual no puede utilizarse como nodo sucesor).

S103: El nodo primario conmuta, en función del nodo sucesor, una vista actual a una vista siguiente que utiliza el nodo sucesor como nodo primario sucesor, de modo que el nodo primario sucesor inicia un consenso en la siguiente vista.

Después de determinarse el nodo sucesor, el nodo primario realiza la conmutación de vista. De una manera existente en que un nodo de respaldo inicia un consenso de conmutación de vista, un proceso en el que el nodo de respaldo inicia el consenso de conmutación de vista puede considerarse un proceso de "destitución" del nodo primario en la vista. A diferencia de la manera existente, en esta implementación de la presente patente, un proceso en el que el nodo primario realiza de forma independiente la conmutación de vista puede considerarse un proceso de "abdicación activa", y el nodo primario realiza la conmutación de vista sin iniciar un consenso. Evidentemente, también se impide un proceso de consenso adicional. Después de realizar la conmutación de vista, un nodo primario recién designado se encarga de iniciar un consenso en la vista conmutada, y puede entenderse que el nodo primario recién designado también realiza el proceso de conmutación de vista anterior. Los detalles se omiten aquí por simplicidad.

De acuerdo con las etapas anteriores, en cualquier vista, el nodo primario supervisa activamente la activación de la condición de cambio de vista. Si se activa la condición de conmutación de vista, el nodo primario debe realizar la conmutación de vista. Además, el nodo primario selecciona un nodo sucesor de entre otros nodos como nodo primario en la siguiente vista. Por consiguiente, el nodo primario realiza la conmutación de vista. En la vista conmutada, el nodo sucesor se utiliza como nuevo nodo primario para procesar un servicio. Además, la conmutación de vista se sigue realizando en función del proceso anterior. Evidentemente, el nodo primario inicia la conmutación de vista anterior. De esta manera se evita que un nodo de respaldo inicie un consenso de conmutación de vista. En otras palabras, puede evitarse un consenso adicional. Por tanto, se puede reducir una cantidad de cálculo adicional y el consumo de tiempo de procesamiento en una red de cadena de bloques.

En la práctica, hay diferentes condiciones de conmutación de vista. A continuación, se describe en detalle la activación de la condición de conmutación de vista.

Primer escenario:

En la práctica, un dispositivo cliente envía una solicitud de servicio a un nodo primario. En un estado normal, después de recibir la solicitud de servicio, el nodo primario difunde la solicitud de servicio a nodos de respaldo en una vista para realizar un consenso sobre la solicitud de servicio. Sin embargo, el nodo primario puede ser un nodo anómalo y no difunde la solicitud de servicio durante mucho tiempo después de recibir la solicitud de servicio. De este modo, el nodo de respaldo inicia un consenso de conmutación de vista. Por lo tanto, para evitar un consenso de conmutación de vista iniciado por el nodo de respaldo debido a que el nodo primario no difunde una solicitud de servicio dentro de un período de tiempo predeterminado, el nodo primario realiza de forma independiente la temporización y supervisa activamente un fenómeno de expiración del nodo primario.

En otras palabras, en este escenario, si la condición de conmutación de vista es que el nodo primario no difunde una solicitud de servicio dentro de un período de tiempo predeterminado, activar la condición de conmutación de vista incluye lo siguiente: recibir, mediante el nodo primario, una solicitud de servicio y no iniciar un consenso sobre la solicitud de servicio dentro de un tiempo predeterminado.

En una operación real, la temporización puede implementarse mediante un programa o un servicio que tiene una función de temporización en el nodo primario, por ejemplo, el temporizador mencionado anteriormente. La temporización se puede iniciar desde un momento en el que el nodo primario recibe la solicitud de servicio. El tiempo predeterminado puede establecerse en 5 s, 10 s, etc., puede determinarse en base a un requisito en la práctica y no constituye limitaciones en la presente patente.

Segundo escenario:

A diferencia del escenario anterior, en este escenario, después de recibir una solicitud de servicio, el nodo primario difunde la solicitud de servicio a nodos de respaldo en una vista actual. En otras palabras, el nodo primario ha iniciado un consenso sobre la solicitud de servicio antes de que expire el tiempo predeterminado. De manera correspondiente, los nodos en la vista realizan un consenso sobre la solicitud de servicio y generan un resultado de consenso.

Cabe destacar que, en este documento, en base a un mecanismo de conmutación de vista existente, si un resultado de consenso es que un consenso falla, el nodo de respaldo inicia un consenso de conmutación de vista. Evidentemente, que un consenso falle puede considerarse una condición de cambio de vista. En otras palabras, en este escenario, cuando el consenso falla, el nodo primario realiza activamente el cambio de vista para evitar que el nodo de respaldo inicie un consenso de cambio de vista adicional.

Además, en base al mecanismo de cambio de vista existente, si un resultado de consenso es que se alcanza un consenso, el nodo primario procede a iniciar un consenso sobre otra solicitud de servicio. Sin embargo, el nodo primario puede estar defectuoso en un proceso de ejecución posterior. Una vez que el nodo primario se vuelve defectuoso, el nodo de copia de seguridad aún inicia un consenso de conmutación de vista. Por lo tanto, para evitar este caso, en esta implementación de la presente patente, después de alcanzarse un consenso, el nodo primario todavía realiza la conmutación de vista.

Se puede observar que, en este escenario, independientemente de si el resultado de consenso es que se alcanza un consenso o que un consenso falla, el nodo primario realiza la conmutación de vista después de determinar el resultado de consenso. En otras palabras, activar la condición de conmutación de vista incluye lo siguiente: recibir, mediante el nodo primario, una solicitud de servicio, iniciar un consenso sobre la solicitud de servicio y determinar un resultado de consenso.

Es decir, en este escenario, el nodo primario necesita determinar que se ha alcanzado un consenso o que un consenso falla. A continuación, se describe en detalle cómo el nodo primario determina que se ha alcanzado un consenso o que un consenso falla.

En primer lugar, cabe destacar que un proceso de consenso basado en una solicitud de servicio es esencialmente un proceso de consenso basado en un protocolo de tres fases. Las tres fases incluyen una fase de preparación previa, una fase de preparación y una fase de compromiso, y forman un proceso de consenso completo. En cada fase, los nodos (incluidos tanto un nodo primario como nodos de respaldo) envían mensajes de consenso entre sí. Es decir, para cada nodo en la vista, la entrada en diferentes fases debe ser reconocida o aceptada por otros nodos. Por lo tanto, cada una de las tres fases puede considerarse un proceso de consenso. Por lo general, cuando todos los nodos entran en una fase de compromiso, se puede considerar que el proceso de consenso se ha completado.

La FIG. 2 es un proceso de consenso de nodos basado en un protocolo de tres fases en una vista. En la FIG. 2, un dispositivo cliente inicia una solicitud de servicio a un nodo numerado como 0 (réplica 0, es decir, un nodo primario), y el nodo primario difunde la solicitud de servicio a nodos de respaldo (réplica 1, réplica 2 y réplica 3), y comienza a realizar un consenso de tres fases. Los nodos que alcanzan un consenso procesan la solicitud de servicio y retroalimentan un resultado de procesamiento al dispositivo cliente.

En consecuencia, se realizan las siguientes operaciones.

1. El nodo primario determina que un consenso falla.

En esta implementación de la presente patente, que un consenso falle se representa como la expiración del proceso de consenso (lo que se denomina simplemente a continuación expiración de consenso, donde la expiración de consenso significa que el tiempo consumido por un proceso de consenso excede una duración de consenso predeterminada, y el tiempo consumido por el proceso de consenso puede contarse desde el momento en el que el nodo primario inicia el consenso). Las razones son las siguientes.

En un caso, el nodo primario es un nodo defectuoso (si un nodo está defectuoso, se puede considerar que los datos utilizados para realizar un consenso en el nodo están defectuosos o la lógica de consenso en el nodo está defectuosa). Es decir, las solicitudes de servicio enviadas por el nodo primario a los nodos de respaldo pueden incluir datos incorrectos (por ejemplo, un número de secuencia de solicitud de servicio incorrecto). Con las garantías del mecanismo BPFT, los nodos de respaldo comprueban las solicitudes de servicio difundidas por el nodo primario. Una vez que los datos incorrectos se incluyen en una solicitud de servicio, un nodo de respaldo normal no reconoce ni acepta la solicitud de servicio. En este caso, el nodo primario puede repetir el proceso de envío de solicitudes de servicio y, en consecuencia, el consenso expira.

De forma alternativa, en otro caso, el nodo primario también es un nodo defectuoso. En este caso, el nodo primario puede enviar un mensaje de notificación incorrecto que indica una fase de consenso a otros nodos de respaldo, es decir, el nodo primario considera "incorrectamente" que se ha entrado en una fase determinada. En este caso, los nodos de respaldo realizan un consenso sobre el mensaje de notificación del nodo primario para confirmar la autenticidad del mensaje de notificación del nodo primario. Del mismo modo, un nodo de respaldo normal todavía no reconoce ni acepta un mensaje de notificación enviado por el nodo primario. En este caso, el nodo primario puede repetir un proceso de envío de un mensaje de notificación incorrecto y, en consecuencia, el consenso expira.

Ciertamente, el contenido anterior es sólo dos posibles casos que pueden dar lugar a la expiración del consenso en la práctica, y no debe considerarse como limitaciones en la presente patente. Evidentemente, del contenido anterior se desprende que, una vez que un consenso expira, el consenso falla.

Por lo tanto, en esta implementación de la presente patente, el nodo primario puede detectar si un consenso falla supervisando una manera prolongada general de un proceso de consenso. Una vez que expira el proceso de consenso, el nodo primario inicia inmediatamente una operación de conmutación de vista, y se puede evitar un proceso de consenso adicional de conmutación de vista iniciado por un nodo de respaldo. Es decir, en esta implementación de la presente patente, un proceso en que el nodo primario determina que el consenso falla puede ser como sigue: El nodo primario supervisa el tiempo consumido por un proceso de consenso desde el momento en el que se inicia un consenso sobre la solicitud de servicio entre los nodos de respaldo en la vista, y determina que el consenso falla cuando se detecta que el tiempo consumido por el consenso excede un tiempo predeterminado.

2. El nodo primario determina que se alcanza un consenso.

Del protocolo de tres fases anterior se desprende que si un nodo entra en una fase de compromiso, el nodo puede procesar una solicitud de servicio y retroalimentar un resultado de procesamiento generado a un dispositivo cliente. Además, para entrar en una fase, cada nodo necesita ser reconocido o aceptado por otros nodos en una vista. Por lo tanto, si un nodo entra en una fase de compromiso, esto indica que el nodo es reconocido o aceptado por otros nodos. Se puede observar que, si el nodo primario entra en la fase de compromiso, esto indica que se ha alcanzado el consenso. Esto se debe a que en el mecanismo PBFT, si un nodo entra en una fase, esto indica que el estado del nodo es reconocido o aceptado por la mayoría de los nodos en la vista. En consecuencia, esto indica que la mayoría de los nodos son nodos correctos.

Por lo tanto, de esta manera, un proceso en que el nodo primario determina que se alcanza un consenso puede ser el siguiente: El nodo primario supervisa una fase correspondiente del nodo primario, y cuando el nodo primario detecta que el nodo primario entra en una fase de compromiso y la duración de consenso predeterminada no expira, determina que el consenso se ha completado. Es decir, cuando se confirma que el nodo primario entra en la fase de compromiso, el nodo primario necesita además cerciorarse de que el tiempo consumido por el nodo primario para entrar en la fase de compromiso no exceda el tiempo predeterminado.

De otra manera en esta implementación de la presente patente, el nodo primario puede no enviar un mensaje de notificación a otros nodos (es decir, el nodo primario puede ser un nodo defectuoso). Sin embargo, el nodo primario aún puede recibir un mensaje de notificación enviado por un nodo de respaldo. En este caso, si una cantidad predeterminada de nodos entra en una fase de compromiso, se puede considerar que el consenso se ha completado.

En la práctica, después de que un nodo entre en una fase de compromiso, el nodo envía generalmente mensajes de notificación a otros nodos en la vista, donde el mensaje de notificación puede ser, por ejemplo, <compromiso, v, n, D(m)>, donde "compromiso" indica que el nodo ha entrado en una fase de compromiso, "v" indica un número de vista, "n" indica un número de secuencia de una solicitud de servicio y "D(m)" indica una firma realizada en la solicitud de servicio por el nodo que envía el mensaje de notificación.

El nodo primario puede recopilar estadísticas sobre los mensajes de notificación que recibe el nodo primario y que indican que se ha entrado en una fase de compromiso. Si una cantidad de mensajes de notificación recibidos es mayor que 2f+1, esto indica que suficientes nodos alcanzan un consenso. Entonces, esto indica que se ha completado el consenso; f es la cantidad máxima de nodos incorrectos tolerables en el mecanismo PBFT. En este caso, el nodo primario puede determinar que el consenso se ha completado.

Por lo tanto, un proceso en que el nodo primario determina que se alcanza un consenso puede ser también el siguiente: El nodo primario supervisa un mensaje de notificación que es recibido por el nodo primario y que indica que el nodo de respaldo entra en una fase de compromiso, y cuando el nodo primario detecta que una cantidad de mensajes de notificación recibidos excede una cantidad predeterminada y que la duración de consenso predeterminada no expira, determina que el consenso se ha completado.

Una vez completado el consenso, el nodo primario inicia la conmutación de vista para cambiar el nodo primario y entrar en la nueva vista.

A continuación, se describe un proceso de conmutación de vista en esta implementación de la presente patente.

En el mecanismo PBFT, cada vista tiene un número correspondiente. Por ejemplo, v en el ejemplo anterior representa el número de la vista actual. En consecuencia, cada nodo de la red de cadena de bloques tiene un número correspondiente. Si hay un total de R nodos en la red de cadena de bloques, los números de los nodos son 0 a R-1, por ejemplo, réplica 0, réplica 1,... y réplica R-1. Existe una relación entre un número de nodo y un número de vista. Si se utiliza la réplica p para representar el nodo con el número p, el número de nodo y un número de vista satisfacen la siguiente ecuación: p=v mod R, donde v es un número entero de 0 a infinito positivo.

Esta relación indica que el número de nodo p se obtiene después del número de vista v módulo la cantidad R de nodos incluidos en una red de cadena de bloques.

En otras palabras, debido a que v varía de 0 a R-1, se garantiza que la identidad del nodo primario se entregue secuencialmente a diferentes nodos. Por ejemplo, si el nodo primario de la vista actual es la réplica 0 (que corresponde al número de vista 0), el nodo primario en la siguiente vista (numerado como 1) es la réplica 1. Todos los nodos se recorren de esta manera.

Por lo tanto, se puede observar que, en esta implementación de la presente patente, un proceso de conmutación de vista incluye lo siguiente: determinar, mediante el nodo primario, un número del nodo primario; determinar, en base al número del nodo primario, un nodo cuyo número esté dispuesto después del número del nodo primario; generar un mensaje de notificación de conmutación de vista en base al nodo determinado; y enviar el mensaje de conmutación de vista a cada nodo de respaldo para realizar la conmutación de vista, de modo que el nodo determinado se convierta en un nodo primario en una siguiente vista.

A continuación, se utiliza una instancia de aplicación específica para la descripción. Como se muestra en la FIG. 3, la instancia incluye las siguientes etapas:

S301: El nodo primario p en una vista numerada como v recibe una solicitud de servicio enviada por un dispositivo cliente y realiza la temporización.

S302: Cuando expira un tiempo predeterminado, determinar si se debe iniciar un consenso sobre la solicitud de servicio a cada nodo de respaldo en la vista; en caso afirmativo, realizar la etapa S303; y de lo contrario, realizar la etapa S305.

S303: Obtener un resultado de consenso.

S304: Determinar si se alcanza el consenso y realizar la etapa S305.

S305: Conmutar la vista v a la vista v+1 y determinar un nodo numerado p+1 como nodo primario en la vista v+1.

El nodo primario inicia la conmutación de vista anterior. De esta manera se evita que un nodo de respaldo inicie un consenso de conmutación de vista.

Lo anterior es un procedimiento de consenso proporcionado en la implementación de la presente patente. En base a la misma idea, las implementaciones de la presente patente proporcionan además un aparato de consenso. Como se muestra en la FIG. 4, para cualquier vista, el aparato de consenso incluye lo siguiente: un módulo de supervisión 401, configurado para supervisar la activación de una condición de conmutación de vista; un módulo de determinación de nodo 402, configurado para seleccionar un nodo sucesor cuando el módulo de supervisión supervisa la activación de la condición de conmutación de vista; y un módulo de conmutación de vista 403, configurado para conmutar, en base al nodo sucesor, una vista actual a una siguiente vista que utiliza el nodo sucesor como nodo primario de cadena de bloques sucesor, de modo que el nodo primario de cadena de bloques sucesor inicie un consenso en la siguiente vista.

En respuesta a la determinación de que se recibe una solicitud de servicio y no se inicia un consenso sobre la solicitud de servicio dentro de un tiempo predeterminado, el módulo de supervisión 401 determina que se supervisa la activación de la condición de conmutación de vista.

En respuesta a la determinación de que se recibe una solicitud de servicio, se inicia un consenso sobre la solicitud de servicio y se determina un resultado de consenso, el módulo de supervisión 401 determina que se supervisa la activación de la condición de conmutación de vista.

El módulo de determinación de nodo 402 determina una siguiente vista de la vista actual y determina un nodo sucesor que corresponde a la siguiente vista.

El módulo de conmutación de vista 403 conmuta la vista actual a la siguiente vista determinada, donde el nodo sucesor se utiliza como nodo primario en la siguiente vista.

Un nodo en cualquier vista incluye un nodo en una red de consorcio de cadena de bloques o una red privada de cadena de bloques.

En la década de 1990, se puede distinguir claramente si una mejora técnica es una mejora de hardware (por ejemplo, una mejora de la estructura de un circuito, tal como un diodo, un transistor o un conmutador) o una mejora de software (una mejora de un procedimiento de método). Sin embargo, a medida que se desarrollan las tecnologías, las mejoras actuales de muchos procedimientos de métodos pueden considerarse mejoras directas a las estructuras de circuitos de hardware. Un diseñador suele programar un procedimiento de método mejorado en un circuito de hardware para obtener la estructura de circuito de hardware correspondiente. Por lo tanto, un procedimiento de método puede mejorarse utilizando un módulo de entidad de hardware. Por ejemplo, un dispositivo lógico programable (PLD) (por ejemplo, una matriz de puertas programables *in situ* (FPGA)) es un circuito integrado de este tipo, y un usuario determina una función lógica del PLD a través de la programación del dispositivo. El diseñador realiza la programación para "integrar" un sistema digital en un PLD sin solicitar al fabricante de chips que diseñe y produzca un chip de circuito integrado específico de la aplicación. Además, en la actualidad, en lugar de fabricar manualmente un chip de circuito integrado, dicha programación se implementa principalmente mediante el uso de un software de "compilador lógico". El software de compilador lógico es similar al de un compilador de software utilizado para desarrollar y escribir un programa. Es necesario escribir el código original en un determinado lenguaje de programación para su compilación. El lenguaje se conoce como lenguaje de descripción de hardware (HDL). Existen muchos HDL, tal como el Lenguaje Avanzado de Expresiones Booleanas (ABEL), el Lenguaje de Descripción de Hardware de Altera (AHDL), Confluence, el Lenguaje de Programación de la Universidad de Cornell (CUPL), HDCal, el Lenguaje de Descripción de Hardware de Java (JHDL), Lava, Lola, MyHDL, PALASM y el Lenguaje de Descripción de Hardware de Ruby (RHDL). El lenguaje de descripción de hardware de circuito integrado de muy alta velocidad (VHDL) y el Verilog son los más utilizados. Un experto en la técnica también debería entender que un circuito de hardware que implementa un procedimiento de método lógico puede obtenerse fácilmente una vez que el procedimiento de método se programa lógicamente utilizando los diversos lenguajes de descripción de hardware descritos y se programa en un circuito integrado.

Un controlador puede implementarse utilizando cualquier procedimiento apropiado. Por ejemplo, el controlador puede ser un microprocesador o un procesador, o un medio legible por ordenador que almacena código de programa legible por ordenador (tal como software o firmware) que puede ser ejecutado por el microprocesador o el procesador, una puerta lógica, un conmutador, un circuito integrado específico de la aplicación (ASIC), un controlador lógico programable o un microprocesador incorporado. Ejemplos del controlador incluyen, pero no se limitan a los siguientes microprocesadores: ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20 y Silicon Labs C8051F320. El controlador de memoria también puede implementarse como parte de la lógica de control de la memoria. Un experto en la técnica también sabe que, además de implementar el controlador utilizando el código de programa legible por ordenador, la programación lógica puede realizarse en etapas de procedimiento para permitir que el controlador implemente la misma función en formas de la puerta lógica, el conmutador, el circuito integrado específico de la aplicación, el controlador de lógica programable y el microcontrolador incorporado. Por lo tanto, el controlador puede considerarse un componente de hardware, y un aparato configurado para implementar varias funciones en el controlador también puede considerarse una estructura en el componente de hardware. De forma alternativa, el aparato configurado para implementar varias funciones puede incluso considerarse un módulo de software que implementa el procedimiento y una estructura en el componente de hardware.

El sistema, aparato, módulo o unidad ilustrados en las implementaciones anteriores pueden implementarse utilizando un chip informático o una entidad, o pueden implementarse utilizando un producto que tenga una función determinada. Un dispositivo de implementación típico es un ordenador. El ordenador puede ser, por ejemplo, un ordenador personal, un ordenador portátil, un teléfono móvil, un teléfono con cámara, un teléfono inteligente, un asistente digital personal, un reproductor multimedia, un dispositivo de navegación, un dispositivo de correo electrónico, una consola de juegos, una tableta electrónica o un dispositivo ponible, o una combinación de cualquiera de estos dispositivos.

Para facilitar la descripción, el anterior aparato se describe dividiendo las funciones en varias unidades. Ciertamente, cuando se implementa la presente patente, una función de cada unidad puede implementarse en uno o más elementos de software y/o hardware.

Un experto en la técnica debe entender que una implementación de la presente divulgación puede proporcionarse como un procedimiento, un sistema o un producto de programa informático. Por lo tanto, la presente divulgación puede usar una forma de implementaciones sólo de hardware, implementaciones sólo de software o implementaciones con

una combinación de software y hardware. Además, la presente divulgación puede utilizar una forma de producto de programa informático implementado en uno o más medios de almacenamiento utilizables por ordenador (incluyendo, pero sin limitarse a, una memoria de disco, un CD-ROM, una memoria óptica, etc.) que incluyen código de programa utilizable por ordenador.

La presente divulgación se describe con referencia a los diagramas de flujo y/o diagramas de bloques del procedimiento, el dispositivo (sistema) y el producto de programa informático en base a las implementaciones de la presente divulgación. Cabe destacar que las instrucciones de programa informático se pueden utilizar para implementar cada uno de los procesos y/o cada uno de los bloques de los diagramas de flujo y/o los diagramas de bloques, así como una combinación de un proceso y/o un bloque de los diagramas de flujo y/o los diagramas de bloques. Estas instrucciones de programa informático pueden proporcionarse a un ordenador de propósito general, un ordenador dedicado, un procesador integrado o un procesador de otro dispositivo de procesamiento de datos programable para generar una máquina, de modo que las instrucciones ejecutadas por el ordenador o el procesador del otro dispositivo de procesamiento de datos programable generen un aparato para implementar una función específica en uno o más procesos de los diagramas de flujo y/o en uno o más bloques de los diagramas de bloques.

Estas instrucciones de programa informático pueden almacenarse en una memoria legible por ordenador que puede dar instrucciones al ordenador o al otro dispositivo de procesamiento de datos programable para que funcione de una manera específica, de modo que las instrucciones almacenadas en la memoria legible por ordenador generen un artefacto que incluya un aparato con instrucciones. El dispositivo con instrucciones implementa una función específica en uno o más procesos de los diagramas de flujo y/o en uno o más bloques de los diagramas de bloques.

Estas instrucciones de programa informático pueden cargarse en el ordenador o en otro dispositivo programable de procesamiento de datos, de modo que se ejecute una serie de operaciones y etapas en el ordenador o el otro dispositivo programable, generándose así un procesamiento implementado por ordenador. Por lo tanto, las instrucciones ejecutadas en el ordenador o el otro dispositivo programable proporcionan etapas para implementar una función específica en uno o más procesos de los diagramas de flujo y/o en uno o más bloques de los diagramas de bloques.

En una configuración típica, un dispositivo informático incluye uno o más procesadores (CPU), una o más interfaces de entrada/salida, una o más interfaces de red y una o más memorias.

La memoria puede incluir una memoria no persistente, una memoria de acceso aleatorio (RAM), una memoria no volátil, y/u otra forma que se encuentre en un medio legible por ordenador, por ejemplo, una memoria de sólo lectura (ROM) o una memoria flash (flash RAM). La memoria es un ejemplo de medio legible por ordenador.

El medio legible por ordenador incluye medios persistentes, no persistentes, móviles e inamovibles que pueden almacenar información utilizando cualquier procedimiento o tecnología. La información puede ser una instrucción legible por ordenador, una estructura de datos, un módulo de programa u otros datos. Ejemplos de un medio de almacenamiento informático incluyen, pero no se limitan a, una memoria de acceso aleatorio paramétrica (PRAM), una memoria de acceso aleatorio estática (SRAM), una memoria de acceso aleatorio dinámica (DRAM) u otro tipo de memoria de acceso aleatorio (RAM), una memoria de sólo lectura (ROM), una memoria de sólo lectura programable eléctricamente borrable (EEPROM), una memoria flash u otra tecnología de memoria, una memoria de sólo lectura en disco compacto (CD-ROM), un disco versátil digital (DVD) u otro almacenamiento óptico, una cinta magnética de casete, un almacenamiento de cinta magnética/disco magnético u otro dispositivo de almacenamiento magnético. El medio de almacenamiento informático puede utilizarse para almacenar información accesible por el dispositivo informático. En base a la definición en la presente memoria descriptiva, el medio legible por ordenador no incluye medios transitorios legibles por ordenador (medios transitorios) tales como una señal de datos modulada y una portadora.

Cabe destacar además que los términos "incluir", "contener" o cualquier otra variante pretenden abarcar una inclusión no exclusiva de modo que un proceso, un procedimiento, un producto o un dispositivo que incluya una lista de elementos no sólo incluye esos elementos, sino que también incluye otros elementos que no están expresamente enumerados, o incluye además elementos inherentes a un proceso, procedimiento, producto o dispositivo de este tipo. Sin más limitaciones, un elemento precedido por "incluye un/a..." no excluye la existencia de otros elementos idénticos en el proceso, procedimiento, producto o dispositivo que incluye el elemento.

Un experto en la técnica debe entender que una implementación de la presente patente puede proporcionarse como un procedimiento, un sistema o un producto de programa informático. Por lo tanto, la presente patente puede utilizar una forma de implementaciones de sólo hardware, implementaciones de sólo software o implementaciones con una combinación de software y hardware. Además, la presente patente puede utilizar una forma de producto de programa informático implementado en uno o más medios de almacenamiento utilizables por ordenador (incluyendo, pero sin limitarse a, una memoria de disco, un CD-ROM, una memoria óptica, etc.) que incluyen código de programa utilizable por ordenador.

La presente patente puede describirse en el contexto general de instrucciones ejecutables por ordenador, por ejemplo, un módulo de programa. Generalmente, el módulo de programa incluye una rutina, un programa, un objeto, un componente, una estructura de datos, etc. que ejecuta una tarea específica o implementa un tipo de datos abstracto específico. La presente patente también se puede practicar en entornos informáticos distribuidos. En los entornos informáticos distribuidos, las tareas se realizan mediante dispositivos de procesamiento remoto conectados a través de una red de comunicaciones. En un entorno informático distribuido, el módulo de programa puede ubicarse en medios de almacenamiento informáticos tanto locales como remotos, incluidos dispositivos de almacenamiento.

Las implementaciones en la presente memoria descriptiva se describen de forma progresiva. En lo que respecta a partes iguales o similares de las implementaciones, se pueden hacer referencias a las implementaciones. Cada una de las implementaciones se centra en una diferencia con respecto a otras implementaciones. En particular, una implementación de sistema es básicamente similar a una implementación de procedimiento, y por lo tanto, se describe de forma breve. En cuanto a las partes relacionadas, se puede hacer referencia a las descripciones relacionadas en la implementación de procedimiento.

Las implementaciones anteriores son implementaciones de la presente patente y no pretenden limitar la presente patente. Un experto en la técnica puede hacer diversas modificaciones y cambios en la presente patente.

REIVINDICACIONES

1. Un procedimiento para conmutar desde una vista actual para realizar un consenso de cadena de bloques en una red de cadena de bloques a una siguiente vista para realizar un consenso de cadena de bloques en la red de cadena de bloques, en el que

a) en cada vista i) un primer nodo de la red de cadena de bloques se utiliza como nodo primario de cadena de bloques, ii) otros nodos de la red de cadena de bloques se utilizan como nodos de respaldo de cadena de bloques, y iii) el nodo primario de cadena de bloques está configurado para recibir una solicitud de servicio desde un dispositivo cliente y difundir la solicitud de servicio a los nodos de respaldo de cadena de bloques para iniciar un consenso de cadena de bloques sobre la solicitud de servicio, y

b) conmutar desde una vista actual para realizar un consenso de cadena de bloques en una red de cadena de bloques a una siguiente vista para realizar un consenso de cadena de bloques en la red de cadena de bloques comprende conmutar desde un nodo primario actual de cadena de bloques a un nodo primario sucesor de cadena de bloques,

estando el procedimiento **caracterizado por que** comprende:

supervisar (S101), mediante el nodo primario actual de cadena de bloques, una condición de conmutación de vista, donde la condición de conmutación de vista comprende i) que el nodo primario actual de cadena de bloques no difunda una solicitud de servicio recibida dentro de un período predeterminado o ii) que un consenso de cadena de bloques iniciado por el nodo primario actual de cadena de bloques falle dentro del período predeterminado;

en respuesta a la supervisión de la condición de conmutación de vista, seleccionar (S102), mediante el nodo primario actual de cadena de bloques, un nodo sucesor de entre los nodos de respaldo de cadena de bloques; y

conmutar (S103), mediante el nodo primario actual de cadena de bloques y en función del nodo sucesor seleccionado, desde una vista actual para realizar un consenso de cadena de bloques a una siguiente vista para realizar un consenso de cadena de bloques, donde la siguiente vista para realizar un consenso de cadena de bloques utiliza el nodo sucesor seleccionado como nodo primario sucesor de cadena de bloques, donde el nodo de cadena de bloques primario sucesor inicia uno o más consensos de cadena de bloques en la siguiente vista.

2. El procedimiento de acuerdo con la reivindicación 1, en el que la condición de conmutación de vista comprende que el nodo primario actual de cadena de bloques no difunda una solicitud de servicio recibida dentro de un período predeterminado, y en el que supervisar la condición de conmutación de vista comprende:

recibir, mediante el nodo primario actual de cadena de bloques, una solicitud de servicio; y

determinar, mediante el nodo primario actual de cadena de bloques, que no se pudo iniciar un consenso de cadena de bloques sobre la solicitud de servicio dentro del período predeterminado.

3. El procedimiento de acuerdo con la reivindicación 2, en el que determinar que el consenso de cadena de bloques sobre la solicitud de servicio no se inició dentro del período predeterminado comprende determinar que los datos incluidos en la solicitud de servicio comprenden datos incorrectos, donde, opcionalmente, los datos incorrectos comprenden un número de secuencia de solicitud de servicio incorrecto.

4. El procedimiento de acuerdo con la reivindicación 2, en el que determinar que el consenso de cadena de bloques sobre la solicitud de servicio no se inició dentro del período predeterminado comprende determinar que el nodo primario de cadena de bloques envió un mensaje de notificación incorrecto al nodo sucesor, donde el mensaje de notificación incorrecto indica que se ha entrado en una fase de consenso de cadena de bloques particular.

5. El procedimiento de acuerdo con la reivindicación 1, en el que la condición de conmutación de vista comprende que un consenso de cadena de bloques iniciado por el nodo primario actual de cadena de bloques dentro del período predeterminado falló, y en el que supervisar la condición de conmutación de vista comprende:

recibir, mediante el nodo primario actual de cadena de bloques, una solicitud de servicio;

iniciar, mediante el nodo primario actual de cadena de bloques, un consenso de cadena de bloques sobre la solicitud de servicio;

recibir, mediante el nodo primario actual de cadena de bloques, un resultado de consenso de cadena de bloques, donde el resultado de consenso de cadena de bloques se genera mediante nodos de cadena de bloques en la vista actual; y

determinar que el resultado de consenso de cadena de bloques indica que el consenso de cadena de bloques falló.

6. El procedimiento de acuerdo con la reivindicación 1, en el que seleccionar, mediante el nodo primario actual de cadena de bloques, el nodo sucesor de entre los nodos de respaldo de cadena de bloques comprende:

determinar, mediante el nodo primario actual de cadena de bloques, una siguiente vista de la vista actual; y

seleccionar un nodo sucesor que corresponda a la siguiente vista.

7. El procedimiento de acuerdo con la reivindicación 6, en el que conmutar desde la vista actual a la siguiente vista que utiliza el nodo sucesor seleccionado como nodo primario sucesor de cadena de bloques comprende conmutar desde la vista actual a la siguiente vista, donde el nodo sucesor se utiliza como nodo primario de cadena de bloques en la siguiente vista.

8. El procedimiento de acuerdo con una cualquiera de las reivindicaciones 1 a 7, en el que un nodo de cadena de bloques en cualquier vista comprende un nodo en una red de consorcio de cadena de bloques o una red privada de cadena de bloques.

9. Un aparato para conmutar desde una vista actual para realizar un consenso de cadena de bloques en una red de cadena de bloques a una siguiente vista para realizar un consenso de cadena de bloques en la red de cadena de bloques, en el que

a) en cada vista i) un primer nodo de la red de cadena de bloques se utiliza como nodo primario de cadena de bloques, ii) otros nodos de la red de cadena de bloques se utilizan como nodos de respaldo de cadena de bloques, y iii) el nodo primario de cadena de bloques está configurado para recibir una solicitud de servicio desde un dispositivo cliente y difundir la solicitud de servicio a los nodos de respaldo de cadena de bloques para iniciar un consenso de cadena de bloques sobre la solicitud de servicio, y

b) conmutar desde una vista actual para realizar un consenso de cadena de bloques en una red de cadena de bloques a una siguiente vista para realizar un consenso de cadena de bloques en la red de cadena de bloques comprende conmutar desde un nodo primario actual de cadena de bloques a un nodo primario sucesor de cadena de bloques,

estando el aparato **caracterizado por** comprender una pluralidad de módulos configurados para realizar operaciones de acuerdo con el procedimiento de una cualquiera de las reivindicaciones 1 a 8.

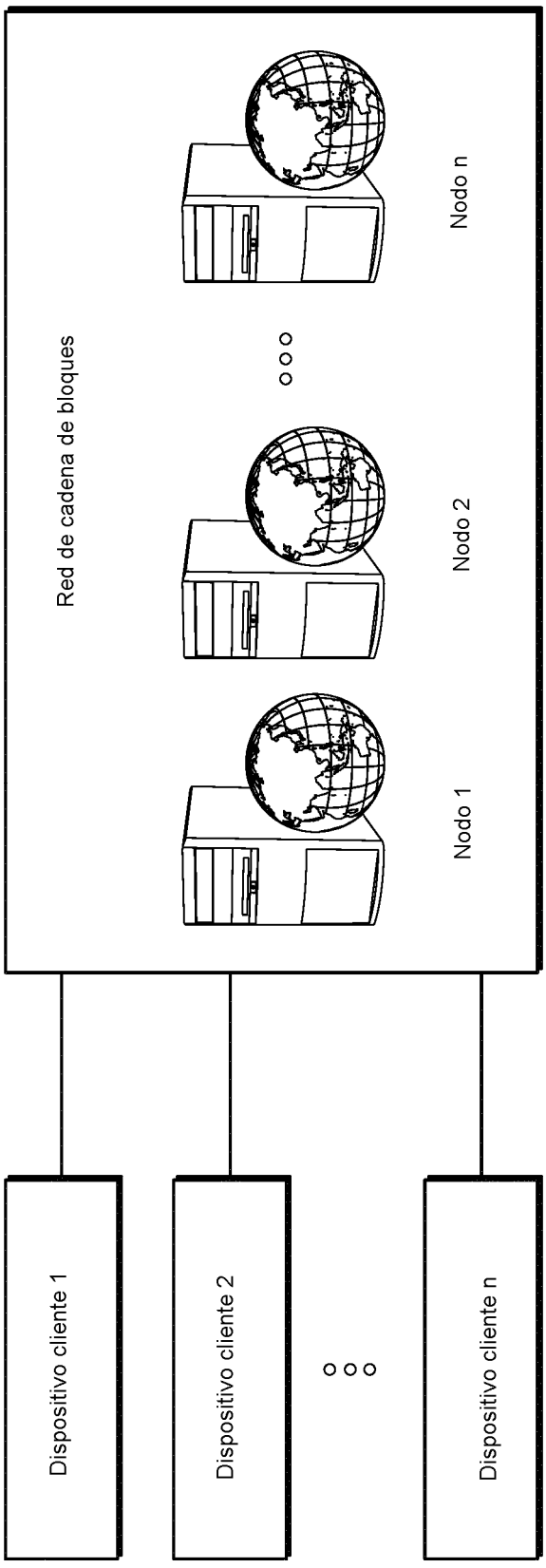
Red de cadena de bloques
Nodo 1
Nodo 2
Nodo n
Dispositivo cliente 1
Dispositivo cliente 2
Dispositivo cliente n

FIG. 1a

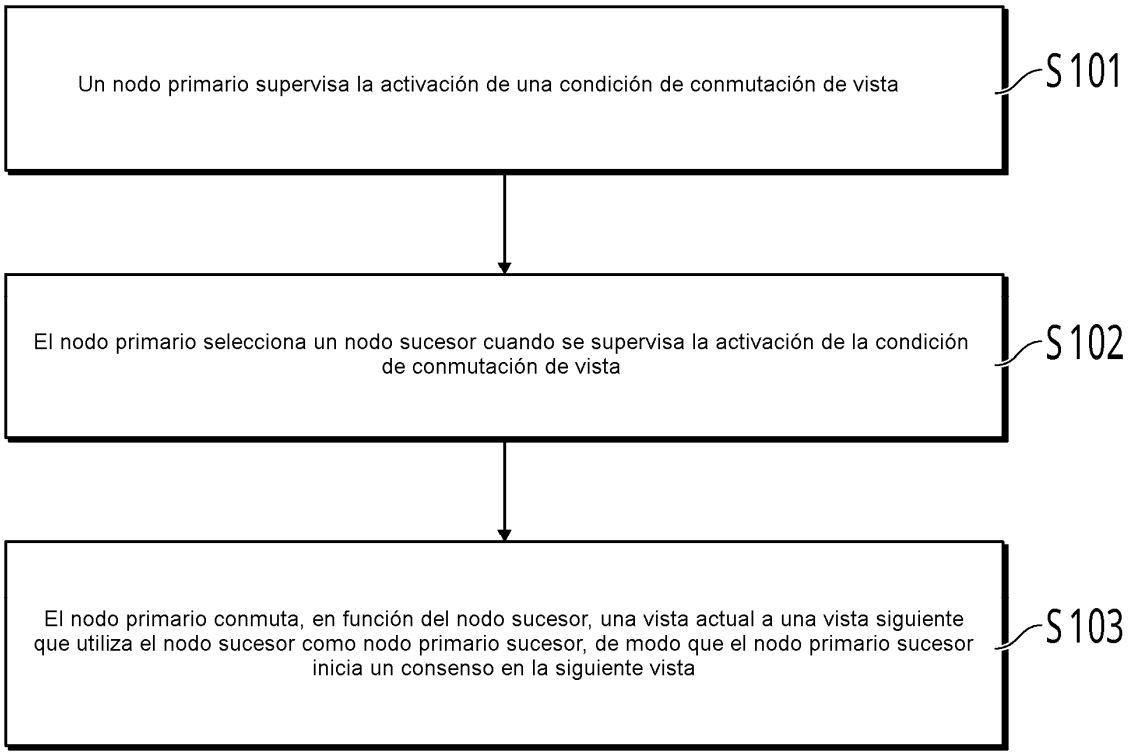
Un nodo primario supervisa la activación de una condición de conmutación de vista
S101
El nodo primario selecciona un nodo sucesor cuando se supervisa la activación de la condición de conmutación de vista
S102
El nodo primario conmuta, en función del nodo sucesor, una vista actual a una vista siguiente que utiliza el nodo sucesor como nodo primario sucesor, de modo que el nodo primario sucesor inicia un consenso en la siguiente vista
S103

FIG. 1b

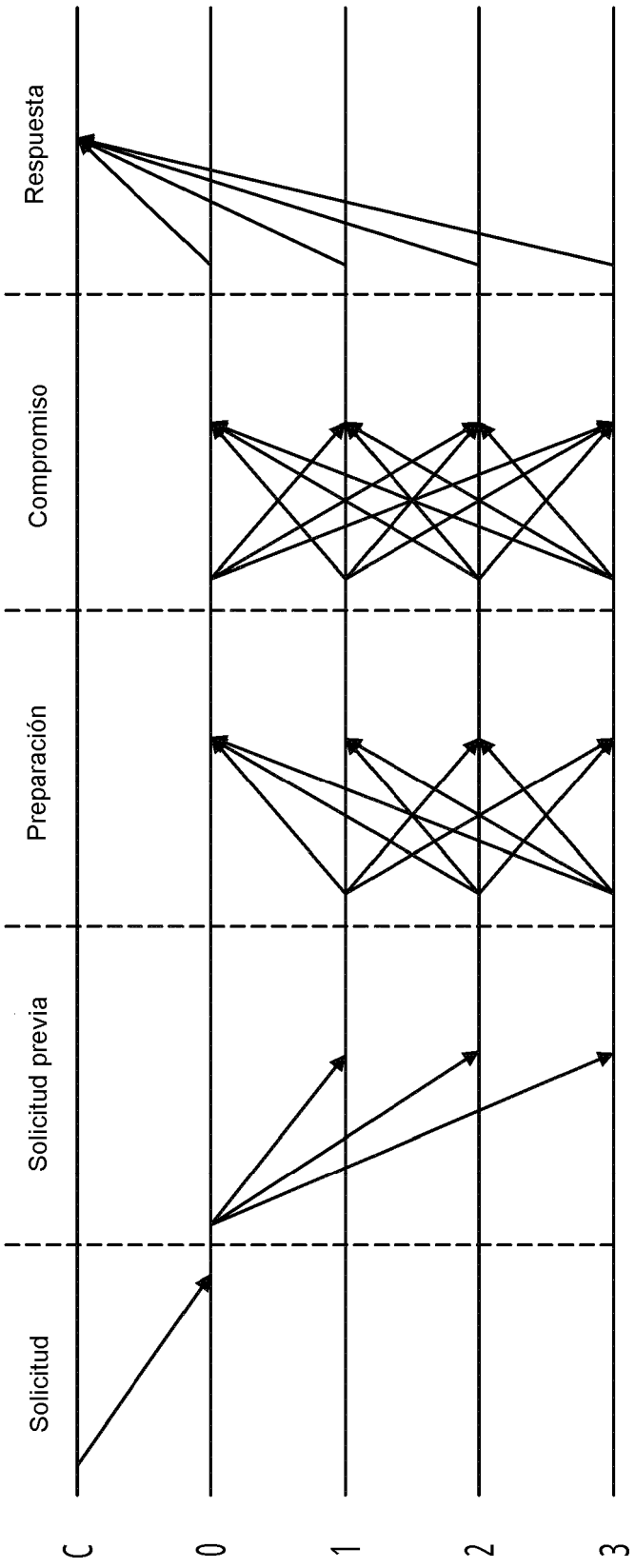
Respuesta
Compromiso
Preparación
Solicitud previa
Solicitud
C
0
1
2
3

FIG. 2

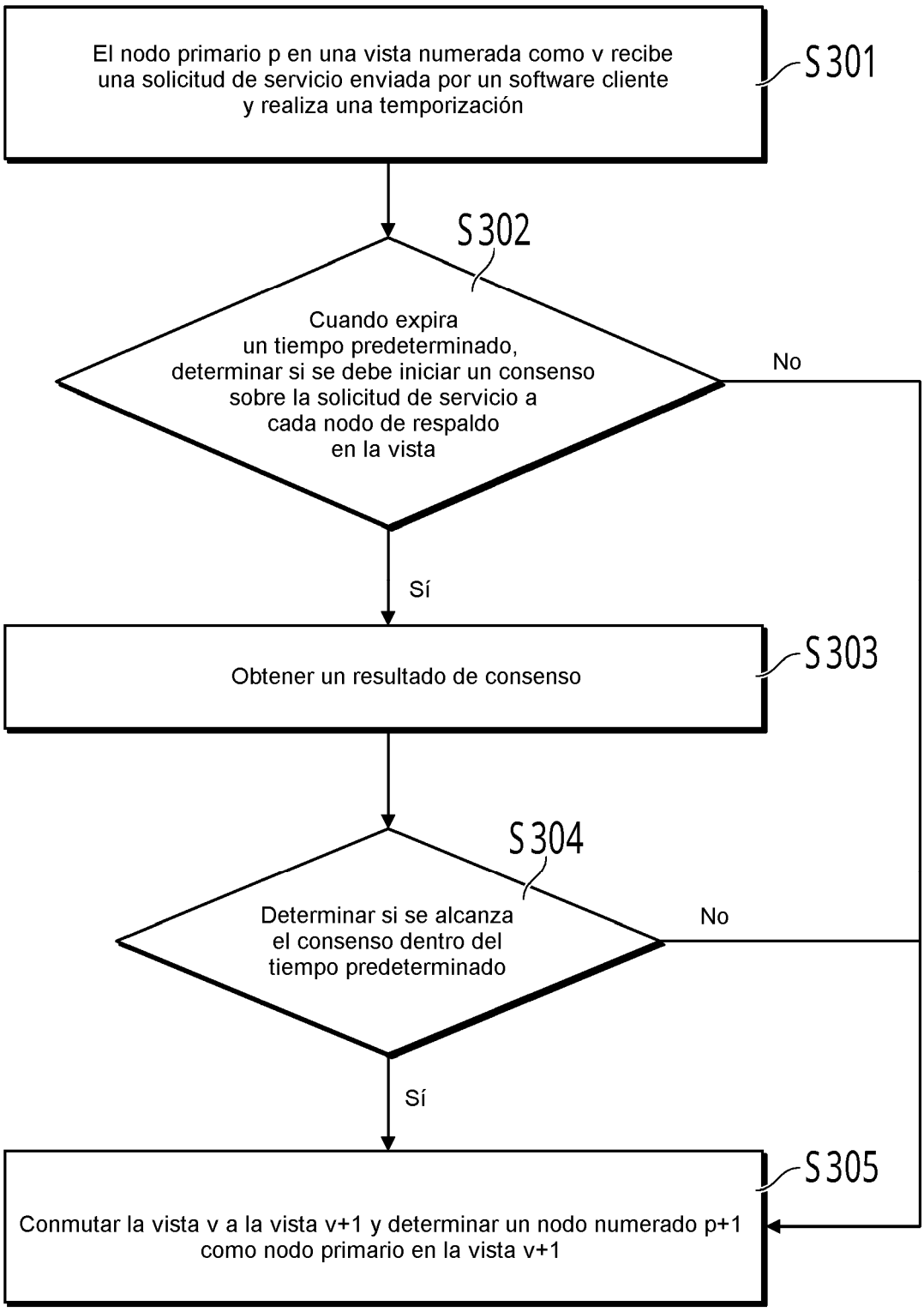
El nodo primario p en una vista numerada como v recibe una solicitud de servicio enviada por un software cliente y realiza una temporización
S301
S302
Cuando expira un tiempo predeterminado, determinar si se debe iniciar un consenso sobre la solicitud de servicio a cada nodo de respaldo en la vista
No
Sí
Obtener un resultado de consenso
S303
S304
Determinar si se alcanza el consenso dentro del tiempo predeterminado
No
Sí
Conmutar la vista v a la vista v+1 y determinar un nodo numerado p+1 como nodo primario en la vista v+1
S305

FIG. 3

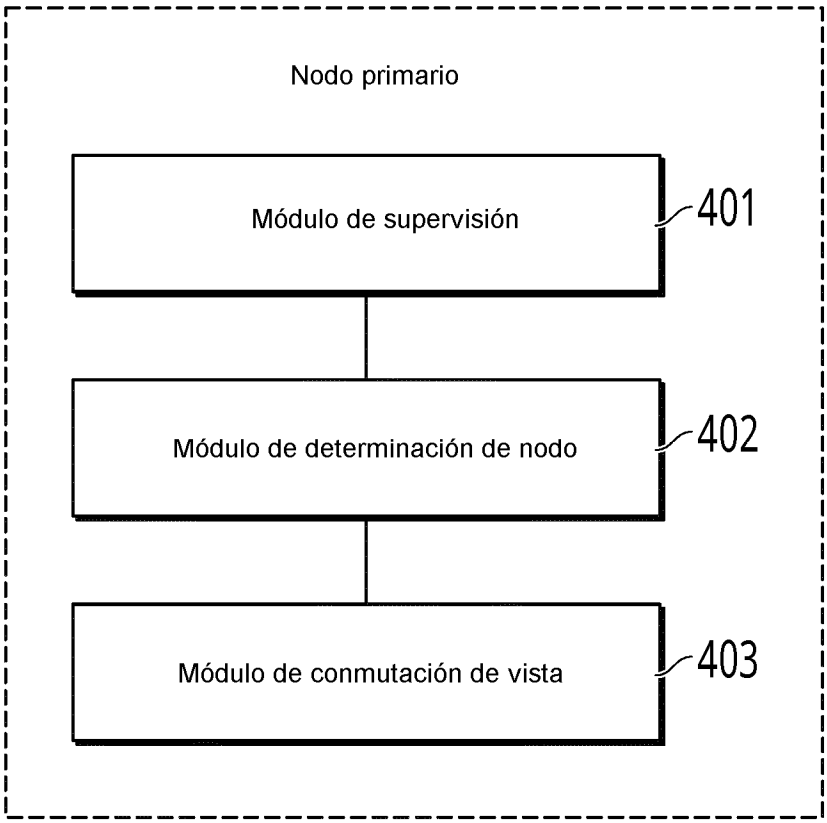
Nodo primario
Módulo de supervisión
401
Módulo de determinación de nodo
402
Módulo de conmutación de vista
403

FIG. 4