

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 768 325**

51 Int. Cl.:

H04W 4/00 (2008.01)

H04L 29/06 (2006.01)

G06F 11/20 (2006.01)

H04L 12/18 (2006.01)

G06Q 50/00 (2012.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **04.10.2013 PCT/US2013/063422**

87 Fecha y número de publicación internacional: **03.07.2014 WO14105247**

96 Fecha de presentación y número de la solicitud europea: **04.10.2013 E 13869434 (4)**

97 Fecha y número de publicación de la concesión europea: **04.12.2019 EP 2939447**

54 Título: **Inducción de un nodo en un grupo**

30 Prioridad:

28.12.2012 US 201261746867 P

15.03.2013 US 201313835888

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

22.06.2020

73 Titular/es:

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

72 Inventor/es:

**AAHLAD, YETURU;
PARKIN, MICHAEL y
AKHTAR, NAEEM**

74 Agente/Representante:

VALLEJO LÓPEZ, Juan Pedro

ES 2 768 325 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Inducción de un nodo en un grupo

5 **Antecedentes**

10 Proyectos colaborativos, que a menudo se facilitan de manera concurrente entre recursos separados globalmente (es decir, proyectos colaborativos de múltiples sitios), se han convertido en lugares comunes para cualquier número de diferentes tipos de proyectos. Ejemplos de tales proyectos incluyen, pero no se limitan a, desarrollo de software, diseño de aviones y diseño de automóviles. Confiar en recursos distribuidos (por ejemplo, recursos en ubicaciones físicamente diferentes, ubicaciones lógicamente diferentes, etc.) para acelerar las líneas de tiempo del proyecto a través de la optimización de la utilización de los recursos humanos y el aprovechamiento de los conjuntos de habilidades de recursos globales ha demostrado que ofrece resultados ventajosos.

15 Una solución informática distribuida utilizada para facilitar un proyecto colaborativo de múltiples sitios se denomina en el presente documento una solución informática colaborativa distribuida de múltiples sitios. Sin embargo, una solución informática colaborativa distribuida en múltiples sitios es solo un ejemplo de una solución informática distribuida. En un ejemplo, una solución informática distribuida comprende una red de ordenadores que operan 811 automóviles. En otro ejemplo, una solución informática distribuida comprende una red de ordenadores en una ubicación geográfica (un centro de datos). En otro ejemplo más, una solución informática distribuida es una pluralidad de ordenadores conectados a un enrutador (es decir, una subred).

25 Si bien existen soluciones informáticas distribuidas convencionales, no están exentas de limitaciones que afectan negativamente a su efectividad, fiabilidad, disponibilidad, escalabilidad, transparencia y/o seguridad. En particular, con respecto a las soluciones informáticas colaborativas distribuidas en sitios múltiples distribuidas convencionales, su capacidad de sincronizar el trabajo desde sitios de desarrollo distribuidos globalmente en tiempo real de manera tolerante a fallos es limitada. Esta incapacidad fuerza cambios en los procedimientos de desarrollo y entrega de software que a menudo causan demoras y aumentan el riesgo. En consecuencia, no se logran por completo los ahorros de costos y las mejoras de productividad que se deben lograr al implementar un proyecto colaborativo utilizando una solución informática distribuida convencional.

35 Las soluciones informáticas colaborativas distribuidas en sitios múltiples distribuidas convencionales obligan indeseablemente a los usuarios a cambiar sus procedimientos de desarrollo. Por ejemplo, soluciones informáticas colaborativas distribuidas en múltiples sitios distribuidas convencionales que carecen de funcionalidades ventajosas asociadas con las capacidades de gestión de información en tiempo real tienen un problema fundamental, ya que no pueden garantizar que los depósitos de sistemas de versiones concurrentes (CVS) locales y remotos estén sincronizados en cualquier momento. Esto significa que existe una gran probabilidad de que los desarrolladores en diferentes sitios puedan sobrescribir o corromper el trabajo de los demás sin darse cuenta. Para evitar tal potencial de sobrescritura y corrupción, estas soluciones informáticas colaborativas distribuidas en sitios múltiples convencionales requieren una ramificación de código fuente excesiva y/o propensa a errores y una fusión manual de archivos para formar parte del proceso de desarrollo. Esto obliga efectivamente a dividir el trabajo de desarrollo en función de las zonas horarias y hace que la colaboración entre equipos de desarrollo distribuidos sea extremadamente difícil, si no imposible.

45 Una máquina de estado replicada es un habilitador preferido de soluciones informáticas distribuidas. Uno de varios ejemplos posibles de una solución informática distribuida es un depósito de información replicado. Por lo tanto, de manera más particular, una máquina de estado replicada es un habilitador preferido de depósitos de información replicada. Una de las varias aplicaciones posibles de los depósitos de información replicada es la distribución de soluciones informáticas colaborativas de múltiples sitios. Por lo tanto, de manera más particular, una máquina de estado replicada es un habilitador preferido de soluciones informáticas colaborativas distribuidas en múltiples sitios.

55 En consecuencia, las soluciones informáticas distribuidas a menudo dependen de máquinas de estado replicadas, depósitos de información replicada o ambos. Las máquinas de estado replicadas y/o los depósitos de información replicados proporcionan generación simultánea, manipulación y gestión de información y, por ello, son aspectos importantes de la mayoría de las soluciones informáticas distribuidas. Sin embargo, los enfoques conocidos para facilitar la replicación de máquinas de estado y facilitar la replicación de depósitos de información no están exentos de inconvenientes.

60 Las implementaciones convencionales para facilitar la replicación de máquinas de estado tienen una o más deficiencias que limitan su efectividad. Una de estas deficiencias es la propensión a la repetida preferencia de los proponentes en un protocolo de acuerdo, que afecta negativamente la escalabilidad. Otra de estas deficiencias es que la implementación de la optimización de líderes débiles requiere la elección de un líder, lo que contribuye a que dicha optimización afecte negativamente la complejidad, velocidad y escalabilidad, y requiere un mensaje más por acuerdo (por ejemplo, 4 en lugar de 3), que afecta negativamente a la velocidad y a la escalabilidad. Otra de estas deficiencias es que los acuerdos deben alcanzarse secuencialmente, lo que afecta negativamente a la velocidad y a la escalabilidad. Otra de estas deficiencias es que la recuperación del almacenamiento persistente es limitada, si no

está ausente por completo, lo que impone una carga considerable en la implementación porque las necesidades de almacenamiento de dicha implementación crecerán continuamente y, potencialmente, sin límite. Otra de estas deficiencias es que la gestión eficiente de grandes propuestas y de un gran número de pequeñas propuestas es limitado, si no está ausente por completo, que afecta negativamente la escalabilidad. Otra de estas deficiencias es que se debe comunicar un número relativamente alto de mensajes para facilitar la replicación de la máquina de estado, que afecta negativamente a la escalabilidad y a la compatibilidad de red de área amplia. Otra limitación es que los retrasos en la comunicación de los mensajes afectan negativamente a la escalabilidad. Otra de estas deficiencias es que abordar los escenarios de fallo al cambiar dinámicamente (por ejemplo, incluir y excluir según sea necesario) a los participantes en la máquina de estado replicada afecta negativamente a la complejidad y a la escalabilidad.

Las implementaciones convencionales para facilitar la replicación de los depósitos de información tienen una o más deficiencias que limitan su efectividad. Uno de estos inconvenientes es que ciertas soluciones informáticas colaborativas de múltiples sitios convencionales requieren un único coordinador central para facilitar la replicación de depósitos de información coordinados centralmente. De manera indeseable, el coordinador central afecta negativamente la escalabilidad porque todas las actualizaciones del depósito de información deben enrutarse a través del coordinador central único. Además, dicha implementación no está altamente disponible porque el fallo del coordinador central único hará que la implementación deje de poder actualizar cualquier réplica del depósito de información. Otro inconveniente es que en una implementación de replicación de depósito de información que se basa en repeticiones de registros, la replicación del depósito de información se facilita de manera activa-pasiva. Por lo tanto, solo una de las réplicas puede actualizarse en un momento dado. Por esto, la utilización de recursos es deficiente porque otras réplicas están inactivas o limitadas para servir a una aplicación de solo lectura, tal como, por ejemplo, Una aplicación de minería de datos. Otra de estas deficiencias resulta cuando la implementación se basa en una replicación débilmente consistente respaldada por heurísticas de resolución de conflictos y/o mecanismos de intervención de aplicaciones. Este tipo de replicación del depósito de información permite actualizaciones conflictivas de las réplicas del depósito de información y requiere una aplicación que use el depósito de información para resolver estos conflictos. Por lo tanto, tal implementación afecta negativamente la transparencia con respecto a la aplicación.

Todavía con referencia a las implementaciones convencionales de facilitar la replicación de los depósitos de información, tienen una o más deficiencias que limitan su efectividad, se sabe que las implementaciones que dependen de una solución de duplicación de disco tienen una o más deficiencias. Este tipo de implementación es una implementación activa-pasiva. Por lo tanto, uno de estos inconvenientes es que la aplicación solo puede usar una de las réplicas en un momento dado. Por esto, la utilización de recursos es pobre porque las otras réplicas (es decir, los espejos pasivos) no son legibles ni escribibles en su función de espejos pasivos. Otro inconveniente de esta implementación particular es que el método de replicación no conoce los límites de transacción de la aplicación. Por esto, en el punto de un fallo, el espejo puede tener un resultado parcial de una transacción y, por lo tanto, puede ser inutilizable. Otra de estas deficiencias es que el método de replicación propaga cambios en la información desde el nodo en el que se originó el cambio a todos los demás nodos. Debido a que el tamaño de los cambios en la información a menudo es mucho mayor que el tamaño del comando que causó el cambio, tal implementación puede requerir una cantidad de ancho de banda indeseablemente grande. Otra de estas deficiencias es que, si la información en el depósito maestro se corrompiera por alguna razón, esa corrupción se propagaría a todas las demás réplicas del depósito. Por esto, el depósito de información puede no ser recuperable o tener que recuperarse de una copia de seguridad anterior, lo que implica una mayor pérdida de información.

El documento US 2008/134052 se refiere a un dispositivo móvil, sistema y método dirigidos a permitir que un miembro de una red social arranque la membresía de otra persona a una aplicación de red social y envíe una invitación utilizando un mecanismo de distribución de igual a igual aumentado. El miembro puede iniciar la membresía proporcionando información sobre el invitado. Un servidor puede determinar la información de redes sociales que se puede compartir, para incluir dentro de la invitación, como información personal sobre el miembro y/o sobre el invitado. La invitación se modifica automáticamente para que parezca que fue enviada por el miembro, falsificando un identificador de la fuente de la invitación. Un enlace u otra información asociada con un enlace a la aplicación de red social que se puede compartir se puede incluir automáticamente en la invitación, donde el enlace está configurado para dirigir al invitado a una configuración de la aplicación de red social apropiada para el dispositivo móvil del invitado.

El documento US 2011/314163 describe un módulo de comunicaciones y un dispositivo de consumo que comprende el modelo de comunicaciones. El módulo de comunicaciones incluye un transceptor inalámbrico para comunicación con la red, por ejemplo, de acuerdo con el protocolo ZigBee, y una interfaz para comunicarse con un procesador central de un dispositivo de consumo o aplicación "inteligente". El módulo recibe datos de eventos programados a través de un enlace inalámbrico en nombre del procesador central y programa eventos para que el procesador central los ejecute al recibir los comandos transmitidos por el módulo. El módulo de comunicaciones puede incluir un módulo central virtual, que traduce comandos entre un protocolo utilizado por el procesador central y un protocolo utilizado por el módulo de comunicaciones. El módulo de comunicaciones también está configurado para buscar y unirse automáticamente a una red, y para descubrir y unirse a los servicios proporcionados a través de la red.

Sumario de la invención

Según los aspectos primero y segundo de la invención, se proporciona respectivamente un método implementado por ordenador para un nodo inductor de acuerdo con la reivindicación 1 y un método implementado por ordenador para un nodo inductor de acuerdo con la reivindicación 8.

De acuerdo con los aspectos tercero y cuarto de la invención, se proporciona respectivamente un medio de almacenamiento de datos tangibles que almacena datos e instrucciones que configuran un dispositivo informático como un nodo inductor de acuerdo con la reivindicación 7 y un medio de almacenamiento de datos tangibles que almacena datos e instrucciones que configuran un dispositivo informático como un nodo inducido de acuerdo con la reivindicación 15.

De acuerdo con los aspectos quinto y sexto de la invención, se proporciona respectivamente un dispositivo informático de acuerdo con las reivindicaciones 6 y 14.

Por lo tanto, una máquina de estado replicada que supera los inconvenientes asociados con las máquinas de estado replicadas convencionales es útil y ventajosa. De manera más específica, un depósito de información replicado construido utilizando una máquina de estado replicada de este tipo sería superior a un depósito de información replicada convencional. Aún más específicamente, un depósito CVS replicado construido usando una máquina de estado replicada de este tipo sería superior a un depósito CVS replicado convencional.

El uso de soluciones informáticas distribuidas como las descritas anteriormente, por lo tanto, ha sido un facilitador clave de tales proyectos de colaboración, ya que proporciona un medio relativamente efectivo y eficiente de compartir información entre ubicaciones separadas físicamente, ubicaciones lógicamente separadas, etc. En cada una de esas ubicaciones, puede haber uno o más nodos informáticos del sistema informático distribuido. Un nuevo nodo, para participar en el proyecto colaborativo, debe invitarse a unirse a los nodos existentes y debe informarse sobre las ubicaciones y los nodos que deben ser visibles para el mismo y con quién el nodo recién invitado puede intercambiar mensajes e interactuar.

Breve descripción de los dibujos

- La figura 1 es un diagrama de bloques que muestra las relaciones funcionales de elementos dentro de una arquitectura de sistema informático de múltiples sitios de acuerdo con una realización.
- La figura 2 es un diagrama de bloques de alto nivel que muestra el despliegue de elementos que componen una arquitectura de sistema informático de múltiples sitios de acuerdo con una realización.
- La figura 3 es un diagrama de bloques que muestra componentes funcionales de una máquina de estado replicada de acuerdo con una realización.
- La figura 4 es un diagrama de bloques que muestra una propuesta emitida por un nodo de aplicación local de acuerdo con una realización.
- La figura 5 es un diagrama de bloques que muestra la estructura de entrada de un secuenciador global de la máquina de estado replicada de la figura 3.
- La figura 6 es un diagrama de bloques que muestra la estructura de entrada de un secuenciador local de la máquina de estado replicada de la figura 3.
- La figura 7 es un diagrama de bloques que muestra un replicador de acuerdo con una realización.
- La figura 8 es un diagrama de bloques de nivel detallado que muestra el despliegue de elementos que componen una arquitectura de sistema informático de múltiples sitios de acuerdo con una realización.
- La figura 9 es un diagrama que muestra aspectos de los dispositivos, métodos y sistemas que permiten una inducción segura y autorizada de un nodo en un grupo de nodos de acuerdo con una realización.
- La figura 10 es un diagrama de bloques de un dispositivo informático con el que se pueden realizar realizaciones.

Descripción detallada

Aquí se describen varios aspectos para facilitar una implementación práctica de una máquina de estado replicada en varias arquitecturas de sistemas informáticos distribuidos (por ejemplo, arquitectura de sistema informático colaborativo distribuido en múltiples sitios). Una persona experta estará al tanto de una o más implementaciones convencionales de una máquina de estado replicada. Por ejemplo, tal implementación convencional de una máquina de estado se describe en la publicación titulada "Implementing fault-tolerant services using the state machine approach: Implementing fault-tolerant services using the state machine approach: A tutorial" (páginas 299-319), escrito por F.B. Schneider, publicado en ACM Computing Surveys 22 en diciembre de 1990 y se incorpora aquí como referencia en su totalidad. Con respecto a la implementación convencional de una máquina de estado en una arquitectura de sistema de aplicaciones distribuidas y como se analiza a continuación en mayor detalle, las realizaciones mejoran aspectos de escalabilidad, fiabilidad, disponibilidad y tolerancia a fallos.

Las realizaciones proporcionan una implementación práctica de una máquina de estado replicada en varias arquitecturas de sistemas informáticos distribuidos (por ejemplo, arquitecturas de sistemas informáticos colaborativos distribuidos en múltiples sitios). De manera más específica, las realizaciones mejoran la escalabilidad,

fiabilidad, disponibilidad y tolerancia a fallos de una máquina de estado replicada y/o un depósito de información replicada en una arquitectura de sistema informático distribuido. En consecuencia, las realizaciones superan ventajosamente una o más deficiencias asociadas con enfoques convencionales para implementar una máquina de estado replicada y/o un depósito de información replicada en una arquitectura de sistema informático distribuido.

5 En una realización, una máquina de estado replicada puede comprender un administrador de propuestas, un administrador de acuerdos, un temporizador de colisión/retroceso y un recuperador de almacenamiento. El administrador de propuestas facilita la gestión de propuestas emitidas por un nodo de una aplicación distribuida para permitir la ejecución coordinada de las propuestas por todos los nodos de la aplicación distribuida que necesitan hacerlo, posiblemente, pero no necesariamente incluyéndose a sí mismo. El administrador de acuerdos facilita el acuerdo sobre las propuestas. El temporizador de colisión/retroceso impide repetidas repeticiones de rondas al intentar llegar a un acuerdo sobre las propuestas. El recuperador de almacenamiento reclama el almacenamiento persistente utilizado para almacenar los acuerdos de propuesta y/o las propuestas.

15 En otra realización, una arquitectura de sistema informático distribuido puede comprender un sistema de red y una pluralidad de sistemas informáticos distribuidos interconectados a través del sistema de red. Cada uno de los sistemas informáticos distribuidos puede incluir una máquina de estado replicada respectiva y un nodo de aplicación local respectivo conectado a la máquina de estado replicada respectiva. La máquina de estado replicada respectiva de cada uno de los sistemas informáticos distribuidos facilita la gestión de propuestas para permitir la ejecución coordinada de las propuestas por el nodo de aplicaciones distribuidas de todos los demás sistemas informáticos distribuidos, facilita el acuerdo sobre las propuestas, impide repetidas anticipaciones de rondas al intentar alcanzar un acuerdo sobre las propuestas y reclama el almacenamiento persistente utilizado para almacenar al menos uno de los acuerdos de propuesta y las propuestas.

25 En otra realización, un método puede comprender una pluralidad de operaciones. Se puede realizar una operación para facilitar el acuerdo sobre las propuestas recibidas de un nodo de aplicación local. Se puede realizar una operación para evitar la repetición de preferencia de rondas al intentar alcanzar un acuerdo sobre las propuestas. Se puede realizar una operación para recuperar el almacenamiento persistente respectivo utilizado para almacenar al menos uno de los acuerdos de propuesta y las propuestas.

30 En al menos una realización, al menos una parte de las propuestas incluye las etapas propuestas correspondientes a la implementación de una actualización de información iniciada por un nodo de una aplicación distribuida. Se puede preservar un orden de emisión de las propuestas mientras se facilita el acuerdo concurrente sobre las propuestas. Una parte de las propuestas puede proponer etapas de escritura correspondientes a una actualización de información respectiva y el administrador de la propuesta puede asignar un número de secuencia local a cada una de las etapas de escritura propuestas y crear un entrelazado globalmente único de las etapas de escritura propuestas, de manera que todos los nodos de una aplicación distribuida que ejecuta las etapas de escritura propuestas ejecuta las etapas de escritura propuestas en una secuencia común. Se puede proporcionar un secuenciador local que incluya una pluralidad de entradas, cada una asociada con una de las propuestas respectivas, como puede ser un secuenciador global que incluye una pluralidad de entradas, cada una de las cuales hace referencia a una de las entradas respectivas del secuenciador local. Cada una de las entradas del secuenciador local puede tener asignado un número de secuencia local único, cada una de las entradas del secuenciador local puede disponerse secuencialmente con respecto al número de secuencia local asignado y, después de que el administrador de acuerdos facilite el acuerdo sobre una de las propuestas, se puede crear una entrada correspondiente a la propuesta sobre la cual se facilita el acuerdo dentro del secuenciador global en respuesta a la determinación de una posición en la que la entrada se coloca dentro del secuenciador global. El recuperador de almacenamiento puede reclamar el almacenamiento persistente eliminando un registro de la propuesta del almacenamiento de la propuesta persistente después de que la posición de la entrada en el secuenciador global sea determinada y conocida por todos los nodos. El temporizador de colisión/retroceso puede configurarse para evitar repetidas preventivas realizando una operación de espera para que pase una duración calculada de retardo de preferencia después de comenzar una de las rondas actuales para un primer proponente antes de iniciar una próxima de las ronda para el primer proponente y/o una operación de espera para que pase una duración calculada del retraso de la ronda de progreso después de comenzar una de las rondas actuales para el primer proponente antes de comenzar una próxima ronda para un segundo proponente.

55 Volviendo ahora a las figuras, la figura 1 muestra una arquitectura de sistema informático de múltiples sitios de acuerdo con una realización (es decir, referida en el presente documento como la arquitectura del sistema informático de múltiples sitios 100) puede incluir una pluralidad de sistemas de aplicación distribuidos 105 interconectados por una Red de Área Amplia (WAN) 110. Cada uno de los sistemas de aplicaciones distribuidas 105 puede incluir una pluralidad de nodos de aplicaciones distribuidas 115 (por ejemplo, un nombre de aplicación en una estación de trabajo), un replicador 120 y una réplica de depósito 125. El replicador 120 de cada sistema de aplicaciones distribuidas 105 puede estar conectado entre la WAN 110, los nodos de aplicaciones distribuidas 115 del respectivo sistema de aplicaciones distribuidas 105 y la réplica de depósito 125 del respectivo sistema de aplicaciones distribuidas 105.

65 En una realización, cada réplica de depósito 125 es un depósito del Sistema de Versiones Concurrentes (CVS). CVS

es un conocido sistema de versiones de código fuente abierto. CVS, como la mayoría de los otros sistemas de versiones de código fuente, está diseñado para ejecutarse como un servidor central en el que múltiples clientes CVS (por ejemplo, unos nodos de aplicaciones distribuidas 115) se conectan utilizando un protocolo CVS sobre, por ejemplo, Protocolo de control de Transmisión (TCP). El servidor CVS, como se ha implementado, bifurca un proceso por conexión de cliente para manejar una solicitud CVS de cada cliente. En consecuencia, el replicador 120 y la réplica de depósito 125 permiten múltiples réplicas de un depósito CVS. Mientras que un depósito de información CVS es un ejemplo de un depósito de información útil con una realización, el objetivo de la presente divulgación es útil para replicar otros tipos de depósitos de información. Las bases de datos y los sistemas de archivos son ejemplos de otros tipos de depósitos de información. En consecuencia, la utilidad y la aplicabilidad de las realizaciones no se limitan a un tipo particular de depósito de información.

Como se analiza a continuación con mayor detalle, cada replicador 120 puede configurarse para escribir actualizaciones de información desde su respectivo sistema de aplicaciones distribuidas 105 en la réplica de depósito 125 de cada otro sistema de aplicaciones distribuidas 105. Cada replicador 120 puede ser el intermediario que actúa como una puerta de enlace de aplicaciones entre clientes CVS (es decir, un respectivo nodo de aplicación distribuido 115) y un servidor CVS dado (es decir, la respectiva réplica de depósito 125). Cada replicador 120 se coordina con otros replicadores pares para garantizar que todas las réplicas de depósito 125 permanezcan sincronizadas entre sí.

A diferencia de las soluciones convencionales, la arquitectura del sistema informático de múltiples sitios 100 no se basa en un coordinador de transacciones central que se sabe que es un punto único de fallo. La arquitectura del sistema informático de múltiples sitios 100 proporciona un enfoque único para la replicación activa-activa en tiempo real, funcionando según el principio de equivalencia de una copia en todas las réplicas de depósito CVS de un sistema de aplicación distribuido. En consecuencia, de acuerdo con una realización, cada réplica del depósito está sincronizada con todas las demás réplicas del depósito en tiempo real, para que los usuarios en cada nodo del sistema de aplicación distribuido (es decir, nodo de aplicaciones distribuidas) siempre están trabajando desde la misma base de información (por ejemplo, programadores que trabajan desde la misma base de código).

Mediante la integración del replicador 120 con la respectiva réplica de depósito 125, cada réplica del depósito se convierte en un nodo activo en la WAN 110 con su propio coordinador de transacciones (es decir, el respectivo replicador 120). Cada coordinador de transacciones distribuidas acepta actualizaciones locales y las propaga a todas las demás réplicas de depósito 125 en tiempo real. En consecuencia, todos los usuarios dentro de la arquitectura del sistema informático de múltiples sitios 100 están trabajando efectivamente desde la misma información de depósito (por ejemplo, un único depósito de información CVS) independientemente de la ubicación. Con este fin, una arquitectura de sistema informático de múltiples sitios de acuerdo con una realización es una solución rentable de gestión de configuración de software (SCM) tolerante a fallos que sincroniza el trabajo de los equipos de desarrollo distribuidos globalmente en tiempo real.

Cuando ocurren fallos en la red o en el servidor, los desarrolladores pueden seguir trabajando. Los cambios se registran en un diario de transacciones del local de los replicadores 120. El diario de transacciones es similar en función a un registro de rehacer de la base de datos. Cuando se restablece la conectividad, el local de los replicadores 120 llega al replicador 120 de otros de los sistemas de aplicación distribuidos 105 para actualizar el local de las réplicas de depósito 125, así como aplicar los cambios capturados en el diario de transacciones local mientras la red o el sistema estaban inactivos. La recuperación puede implementarse automáticamente, sin ninguna intervención de un administrador de CVS. Esta capacidad de autocuración asegura una pérdida cero de datos, no pierde tiempo de desarrollo y elimina el riesgo de error humano en un escenario de recuperación ante desastres.

Los beneficios de trabajar esencialmente con la misma información de depósito incluyen no tener que cambiar los procedimientos de desarrollo cuando el desarrollo se traslada al extranjero, no tener que permanecer inactivo mientras espera que se completen las grandes compilaciones cuando se integra el trabajo desde múltiples sitios, ser capaz de detectar problemas de desarrollo antes y gastar menos recursos (por ejemplo, reducir la utilización de recursos redundantes) en garantía de calidad. Además, la recuperación ante desastres no es un problema porque la capacidad integrada de autocuración proporciona prevención de desastres. El trabajo nunca se pierde cuando un sistema se cae.

Como se divulgó anteriormente, la implementación de una máquina de estado replicada de acuerdo con una realización impacta ventajosamente a la escalabilidad, fiabilidad, disponibilidad y tolerancia a fallos de una máquina de estado replicada. Al afectar ventajosamente a la escalabilidad, fiabilidad, disponibilidad y tolerancia a fallos, el presente proporciona un enfoque práctico para implementar una máquina de estado replicada en una arquitectura de sistema informático de múltiples sitios. Al implementar una máquina de estado replicada de acuerdo con una realización, se cumplirán todos o una parte de los siguientes objetivos: permitir que los nodos de un sistema informático distribuido de ordenadores desarrollen su estado de manera coordinada; permitir que se conserve la consistencia de un sistema distribuido de ordenadores a pesar de fallos arbitrarias o fallos parciales de las redes de ordenadores, ordenadores o recursos informáticos; permitiendo que se cree un sistema fiable de nodos de aplicación distribuidos a partir de componentes con modesta fiabilidad; asegurar la terminación del protocolo de acuerdo con probabilidad en función del tiempo asintóticamente próximo a 1, a pesar de las colisiones en el protocolo del

acuerdo; eliminar colisiones en el protocolo del acuerdo en condiciones normales de operación: mejorar la eficiencia del protocolo del acuerdo; reducir y limitar el uso de la memoria y el disco de la máquina de estado replicada: reducir el uso de los recursos de red por la máquina de estado replicada; aumentar el rendimiento de las transiciones de estado realizables por la máquina de estado replicada; y permitir una gestión más eficiente de los recursos de memoria y disco por parte de los nodos de aplicación distribuidos atendidos por la máquina de estado replicada.

Como se muestra en la figura 2, la funcionalidad de cálculo de múltiples sitios de acuerdo con una realización se ve facilitada por una pluralidad de máquinas de estado replicadas 200 que interactúan entre sí y con un respectivo nodo de aplicación local 205 a través de un sistema de red 210. Preferentemente, pero no necesariamente, cada nodo de aplicación local 205 puede ser el de una aplicación distribuida y sirve como proponente de propuestas o aceptador de propuestas en cualquier momento dado. En una realización, el sistema de red 210 puede incluir una red de área amplia (WAN) conectada entre las máquinas de estado replicadas 200 y una red de área local (LAN) respectiva conectada entre cada máquina de estado replicada 200 y el respectivo nodo de aplicación local 205. Por ejemplo, cada máquina de estado replicada 200 y su respectivo nodo de aplicación local 205 están situados en un sitio respectivo para un proyecto de cálculo colaborativo de múltiples sitios. La porción LAN del sistema de red 210 facilita el intercambio de información a nivel local (es decir, entre cada máquina de estado replicada 200 y su respectivo nodo de aplicación local 205) y la porción WAN del sistema de red 210 facilita el intercambio de información a nivel mundial (es decir, entre las máquinas de estado replicadas 200). Mientras una LAN, una WAN o ambas son ejemplos de componentes constituyentes de un sistema de red de acuerdo con una realización, las realizaciones no se limitan a una configuración particular del sistema de red. Por ejemplo, otras realizaciones de un sistema de red de acuerdo con una realización incluyen un sistema de red ad-hoc que incluye ordenadores integrados en un automóvil, un sistema de red que comprende una pluralidad de subredes en un centro de datos y un sistema de red que incluye una subred dentro de un centro de datos.

La figura 3 es un diagrama de bloques que muestra los componentes funcionales de cada máquina de estado replicada 200 que se muestra en la figura 2. Cada máquina de estado replicada 200 puede incluir un administrador de propuestas 220, almacenamiento de propuestas de persistencia 230, un administrador de acuerdos 240, una tienda de convenios, 245, una capa de Protocolo distribuido de transferencia de archivos (DFTP) 250, un temporizador de colisión y retroceso 260, un secuenciador local 270, un secuenciador global 280 y un recuperador de almacenamiento 290 (es decir, un recolector de basura de almacenamiento persistente). El administrador de propuestas 220, el almacenamiento de propuestas de persistencia 230, el administrador de acuerdos 240, la tienda de acuerdos, 245, la capa DFTP 250, el temporizador de colisión y retroceso 260, el secuenciador local 270, el secuenciador global 280 y el recuperador de almacenamiento 290 están interconectados al menos entre sí para permitir la interacción entre los mismos. Como se verá en la siguiente descripción, cada uno de los componentes funcionales de la máquina de estado replicada admite una funcionalidad ventajosa de acuerdo con una realización.

Gestión de propuestas

Cada nodo de aplicación local 205 propone una secuencia de propuestas a la respectiva máquina de estado replicada 200. La secuencia de propuestas que son propuestas por cada nodo local 6 constituye una secuencia local de ese respectivo nodo local 205, que puede mantenerse dentro del secuenciador local 270 de la respectiva máquina de estado replicada 200. El administrador de propuestas 220 de cada máquina de estado replicada 200 organiza la secuencia respectiva de propuestas en una única secuencia global de propuestas respectiva, que puede mantenerse dentro del secuenciador global 280 de la respectiva máquina de estado replicada 200. Cada secuencia global de propuestas tiene las siguientes propiedades: cada propuesta de cada secuencia local ocurre exactamente una vez en la secuencia global respectiva, el orden relativo de cualquiera de las dos propuestas en una secuencia local puede conservarse opcionalmente en la secuencia global respectiva, y las secuencias globales (con o sin el orden local preservado) asociadas con todos los nodos de aplicación locales 205 son idénticas.

Cuando un hilo del nodo de aplicación local 205 propone una propuesta (por ejemplo, escribir etapas) a la respectiva máquina de estado replicada 200, la máquina de estado replicada 200 asigna un número de secuencia local a la propuesta. Esa máquina de estado replicada 200 determina entonces un número de acuerdo para esa propuesta. Como será evidente por las descripciones a continuación, el número de acuerdos determina la posición de una propuesta respectiva en la secuencia global. La máquina de estado replicada 200 guarda un registro de la propuesta en su almacenamiento de propuesta persistente 230. La máquina de estado replicada 200 luego devuelve el control del subproceso del nodo de aplicación local al nodo de aplicación local, por lo tanto, el subproceso puede estar disponible para su uso por la aplicación local y no estar inactivo mientras se ejecuta el protocolo del acuerdo. La máquina de estado replicada inicia entonces un protocolo de acuerdo para la propuesta a través del administrador de acuerdos 240. Cuando termina el protocolo del acuerdo, la máquina de estado replicada 200 compara el acuerdo preparado por el protocolo de acuerdo con el acuerdo propuesto contenido en la propuesta. Si el acuerdo alcanzado por el administrador de acuerdos 240 puede ser el mismo que el de la propuesta, la máquina de estado replicada 200 concluye el procesamiento de la propuesta. De lo contrario, la máquina de estado replicada 200 intenta repetidamente llegar a un acuerdo sobre la propuesta utilizando un nuevo número de acuerdo hasta que el acuerdo alcanzado por el administrador de acuerdos pueda ser el mismo que el de la propuesta. Tras la conclusión de un acuerdo, cada nodo de aplicación local 205 pone en cola la propuesta ahora acordada en su secuencia global. Posteriormente, cada nodo de aplicación local 205 de la aplicación distribuida pone en cola y ejecuta las propuestas

contenidas dentro de la secuencia global.

La figura 4 muestra una realización de una propuesta de acuerdo con una realización, a la que se hace referencia en el presente documento como la propuesta 300. La propuesta 300 puede incluir un identificador de proponente 320 (es decir, un identificador de un nodo de aplicación local), un número de secuencia local (LSN) 330, un número de secuencia global (GSN) 340, un número de acuerdo 350 y contenido de la propuesta 360. Preferentemente, pero no necesariamente, las propuestas emitidas por cada nodo de aplicación local 205 tienen la estructura de la propuesta 300.

La figura 5 muestra una realización de una secuencia local de acuerdo con una realización, que se denomina en el presente documento como la secuencia local 400. La secuencia local 400 puede incluir el contenido de cada una de las propuestas para el nodo de aplicación local 205 respectivo. De manera más específica, dichos contenidos incluyen el identificador del proponente, el número de secuencia local (LSN), el número de secuencia global (GSN), el número de acuerdo y el contenido de la propuesta. Preferentemente, pero no necesariamente, la secuencia local asociada con cada máquina de estado replicada 200 tiene la estructura de la secuencia local 400.

La figura 6 muestra una realización de una secuencia global de acuerdo con una realización, que se denomina en el presente documento como la secuencia global 500. La secuencia global puede incluir el número de secuencia global para una serie de propuestas y un identificador de secuencia local. En una realización, el identificador de secuencia local puede ser un puntero a la secuencia local respectiva (es decir, según se representa, la secuencia local 400). En otra realización, el identificador de secuencia local puede ser una clave para una tabla de secuencias locales. Preferentemente, pero no necesariamente, la secuencia global asociada con cada máquina de estado replicada 200 tiene la estructura de la secuencia global 500.

Acuerdos concurrentes

Las máquinas de estado replicadas 200 representadas en las figuras 2 y 3, que son máquinas de estado replicadas de acuerdo con una realización, incorporan un mecanismo de acuerdo concurrente que permita el acuerdo sobre múltiples propuestas de un proponente para progresar simultáneamente mientras, opcionalmente, preservar el orden en que el proponente presentó las propuestas. Por el contrario, las máquinas de estado replicadas convencionales intentan llegar a un acuerdo sobre una propuesta después de alcanzar un acuerdo sobre una propuesta anterior. Esta metodología convencional de máquina de estado replicada asegura que una máquina de estado replicada convencional conserve el orden local de propuestas. Por lo tanto, si un proponente primero propone la propuesta A y luego propone la propuesta B, la máquina de estado replicada convencional asegura que la propuesta A sea acordada y antes de la propuesta B. Sin embargo, a diferencia de una máquina de estado replicada que implementa un mecanismo de retroceso de acuerdo con una realización, esta metodología de la convención ralentiza el funcionamiento de la máquina de estado replicada convencional ya que el acuerdo sobre la propuesta B no puede iniciarse hasta que la propuesta A haya llegado a un acuerdo.

Con referencia ahora a aspectos de una realización, cada objeto (es decir, una entrada) en la secuencia global puede enumerarse secuencialmente. El número asociado con un objeto en la secuencia global identifica su posición con respecto a los otros objetos en la secuencia global. Por ejemplo, un objeto numerado 5 precede a un objeto numerado 6 y puede estar precedido por un objeto numerado 4. Además, cada objeto en la secuencia global contiene un identificador para una secuencia local, tal como el identificador de secuencia local 400 que se muestra en la figura 5. Si la solicitud no requiere la preservación del orden de envío (es decir, orden emitida desde la fuente), cada objeto en la secuencia global contiene la propia propuesta. En ese caso, la propuesta puede obtenerse directamente de la secuencia global en lugar de indirectamente a través de la secuencia local. En una de varias realizaciones posibles, el identificador de la secuencia local puede ser un puntero a la secuencia local. En otra realización, el identificador de la secuencia local puede ser una clave para una tabla de secuencias locales.

Refiriéndonos ahora a las figuras 2 y 3, cada secuencia local contiene las propuestas de la máquina de estado replicada 200 propuesta por uno de los proponentes de la máquina de estado replicada 200. Cada nodo de aplicación local 205 de la máquina de estado replicada 200 mantiene una secuencia local para cada uno de los proponentes asociados con la máquina de estado replicada 200. Los objetos en la secuencia local están numerados secuencialmente. El número asociado con un objeto en la secuencia local identifica su posición con respecto a los otros objetos en la secuencia local. Por ejemplo, el objeto numerado 5 precede al objeto numerado 6 y puede estar precedido por el objeto numerado 4. Cada objeto en la secuencia local contiene una propuesta de la máquina de estado replicada 200.

En cada nodo de aplicación local 205 de la máquina de estado replicada 200, después de llegar a un acuerdo sobre una propuesta, la propuesta puede agregarse a la secuencia global. La identidad del proponente (por ejemplo, el ID del proponente 320 en la figura 4) puede usarse como la clave para buscar una secuencia local de la tabla de secuencias locales. El número de secuencia local (LSN) de la propuesta determina la posición de la propuesta en la secuencia local. La propuesta puede entonces insertarse en la posición determinada en la secuencia local. El número de acuerdo de la propuesta (por ejemplo, acuerdo número 350 en la figura 4) determina la posición de la propuesta en la secuencia global. Se puede insertar un identificador para la secuencia local en la posición

determinada en la secuencia global (es decir, en función del número de acuerdo). El GSN es un campo opcional de contabilidad para asociar con la propuesta para designar la posición real de la propuesta en la secuencia global cuando se consume como se describe en el párrafo a continuación.

5 En una realización, un hilo dedicado consume la secuencia global. El hilo espera hasta que se complete la siguiente posición en la secuencia global. El hilo luego extrae la secuencia local almacenada en esa posición de la secuencia global. El hilo entonces espera hasta que se complete la siguiente posición en la secuencia local. El hilo luego extrae la propuesta de la máquina de estado replicada 200 almacenada en esa posición de la secuencia local. Una persona experta apreciará que las propuestas no se extraerán necesariamente de acuerdo con la secuencia de números de
10 acuerdo, pero se extraerá exactamente en la misma secuencia en todos los nodos de la aplicación. Esta secuencia de extracción puede registrarse para la comodidad de la contabilidad en el campo GSN, pero por lo demás no es esencial para la operación de la máquina de estado replicada 200. Por ejemplo, suponga que un nodo de aplicación (A) envía sus dos primeras propuestas a la máquina de estado replicada (LSN 1 y LSN 2). Suponga además que la máquina de estado replicada alcanzó un acuerdo sobre LSN 2 antes de llegar a un acuerdo sobre LSN 1. Por lo tanto, el número de acuerdo para A:1 (LSN 1 del nodo de aplicación A) es 27 y el número de acuerdo para LSN 2 es 26 (es decir, hubo un total de 25 acuerdos anteriores sobre propuestas de otros nodos de aplicación y ningún acuerdo intermedio sobre propuestas de otros nodos de aplicación entre A:1 y A:2). Usando el método anterior, A:1 se extraerá de la secuencia global en la posición 26 y A:2 en la posición 27. Por lo tanto, GSN respetará a LSN mayores, pero el número del acuerdo no necesariamente tiene que hacerlo. Esta metodología permite que una
15 máquina de estado replicada de acuerdo con una realización procese acuerdos de manera concurrente.

El subproceso luego aplica la propuesta de la máquina de estado replicada 200. En una realización, la aplicación de la propuesta puede realizarse invocando una función de devolución de llamada registrada por una aplicación de la máquina de estado replicada 200.

25 Retroceso y prevención de colisiones

Una máquina de estado replicada de acuerdo con una realización (por ejemplo, la máquina de estado replicada 200) puede incluir un mecanismo de retroceso para evitar la preferencia repetida de los proponentes (por ejemplo, nodos de aplicación local 205) en el protocolo de acuerdo del administrador de acuerdos 240. Por el contrario, cuando una ronda iniciada por un primer proponente se adelanta a una ronda iniciada por un segundo proponente, las máquinas de estado replicadas convencionales permiten que el proponente anticipado inicie inmediatamente una nueva ronda con un número de ronda mayor que el del anticipador. De manera indeseable, esta metodología convencional prepara el escenario para las repetidas anticipaciones de rondas, que puede llevar a un protocolo de acuerdo a agitarse durante un tiempo inaceptablemente largo (por ejemplo, perpetuamente).

Al facilitar el retroceso de acuerdo con una realización cuando se anticipa una ronda, el proponente calcula la duración de un retraso de preferencia. El proponente espera entonces esa duración calculada antes de iniciar la siguiente ronda de acuerdo con un algoritmo convencional para iniciar dicha próxima ronda.

Al facilitar la prevención de colisiones de acuerdo con una realización, cuando un primer proponente siente que un segundo proponente ha iniciado una ronda, el primer proponente calcula la duración de una ronda de retraso en el progreso. El primer proponente se abstiene de iniciar una ronda hasta que la duración del retraso calculado haya expirado.

En una realización, un retraso dado crece exponencialmente con las posteriores anticipaciones de una ronda. Además, el retraso es preferiblemente aleatorio.

Existen varios métodos posibles que pueden usarse para determinar la duración de un retraso dado. Una fuente de inspiración para métodos viables es la literatura sobre protocolos de detección de colisión/acceso múltiple con detección de portadoras (CSMA/CD) para Ethernet sin conmutación. Un protocolo CS-MA/CD es un conjunto de reglas que determina cómo responden los dispositivos de red cuando dos dispositivos de red intentan usar un canal de datos simultáneamente.

En una de varias realizaciones posibles, el siguiente método determina la duración de un retraso calculado. Un administrador que implementa la máquina de estado replicada 200 configura cuatro valores numéricos. Para el propósito de la descripción de esta realización, los valores se llaman A, U, R y X. En una configuración válida, el valor R es mayor que cero y menor que uno: el valor A es mayor que cero; el valor X es mayor que uno; el valor U es mayor que el valor A. Se puede estimar el tiempo de ejecución del protocolo de acuerdo. Uno de los varios estimadores posibles del tiempo de ejecución del protocolo de acuerdo puede ser un promedio de ventana móvil de tiempos de ejecución pasados del protocolo de acuerdo. A efectos de esta descripción, este valor estimado se llama E. A se multiplica por U para determinar el valor M. Se selecciona el mayor de los dos valores A y E. A efectos de esta descripción, este valor seleccionado se llama F. F se multiplica por X para determinar el valor C. Se genera un valor aleatorio V a partir de una distribución uniforme entre cero y C por R. Si C es mayor que M. V se resta de C para calcular D. De lo contrario, V se agrega a C para calcular D.

El valor D calculado puede usarse como el retraso de redondeo en progreso. También se puede usar como el retraso preventivo la primera vez que un nodo de aplicación local 205 se adelanta en la ejecución de una instancia de protocolo de acuerdo. Cada vez posterior, el nodo de aplicación local 205 puede ser adelantado en la ejecución de la instancia del protocolo de acuerdo, Se puede calcular un nuevo valor D utilizando el valor anterior D en lugar del valor A en el método anterior. El nuevo valor D puede usarse como el retraso de preferencia.

Reclamación de almacenamiento persistente

Una máquina de estado replicada de acuerdo con una realización (por ejemplo, la máquina de estado replicada 200) reclama el almacenamiento persistente utilizado para garantizar su tolerancia a fallos y alta disponibilidad. Con referencia a las figuras 2 y 3, el recuperador de almacenamiento 290 elimina un registro de una propuesta que se ha propuesto desde el almacén de propuestas 230 después de que la máquina de estado replicada 200 haya determinado la posición de la propuesta que se ha propuesto en la secuencia global y todos los nodos de aplicación estén informados de esta posición. A intervalos periódicos, cada nodo de aplicación local 205 envía un mensaje a los nodos locales 205 indicando la posición más alta contiguamente poblada en su copia de la secuencia global. A intervalos periódicos, el recuperador de almacenamiento 290 elimina todos los acuerdos hasta la posición más alta contiguamente poblada en todas las copias de la secuencia global que ya no son necesarias para el nodo de aplicación local. De esta manera, cada máquina de estado replicada 200 reclama almacenamiento persistente.

Reservas débiles

Una máquina de estado replicada de acuerdo con una realización (por ejemplo, la máquina de estado replicada 200) proporciona un mecanismo opcional de reserva débil para eliminar la preferencia de los proponentes en condiciones normales de operación. Con referencia a las figuras 2 y 3, cada proponente que conduce una respectiva máquina de estado replicada 200 puede estar numerada contiguamente. Por ejemplo, si hay tres proponentes, pueden estar numerados 1, 2 y 3. El número de un proponente determina qué propuestas de la máquina de estado replicada 200 respectiva conducirá un proponente correspondiente. Si el número de un proponente es M, y si hay N proponentes, el proponente dirigirá las propuestas numeradas $M + (k \cdot \text{veces} \cdot N)$ (es decir, M más k multiplicado por N, para todos los valores enteros de k mayores o iguales a 0). Para permitir que un sistema de aplicaciones distribuidas progrese cuando todos los proponentes de dicho sistema no estén disponibles, si una propuesta de la máquina de estado replicada 200 no se puede determinar de manera oportuna, cualquier proponente asociado con la respectiva máquina de estado replicada 200 puede proponer una "no operación" (es decir, no-op) para esa propuesta. Para hacer que esta optimización sea transparente para la aplicación distribuida, la máquina de estado replicada 200 no entrega las propuestas no operativas a la aplicación distribuida. Ninguna operación se refiere a una etapa de cálculo que, en general, no tiene ningún efecto y, en particular, no cambia el estado de la máquina de estado replicada asociada.

Números redondos distinguidos y justos

Una máquina de estado replicada de acuerdo con una realización asegura que uno de una pluralidad de proponentes competidores no se adelantará cuando se use el mismo número redondo para propuestas competidoras. Por el contrario, las máquinas de estado replicadas convencionales no incluyen un mecanismo que garantice que uno de una pluralidad de proponentes competidores no se anticipará cuando se utilice el mismo número redondo para propuestas competidoras. Un número redondo en tales máquinas de estado replicadas convencionales puede ser un valor monótono, lo que hace posible que todos los proponentes sean anticipados.

Además del componente monótono, en una realización, el número redondo puede contener un componente distinguido. En una realización, se puede asociar un número entero pequeño y distinto con cada proponente de cada máquina de estado replicada 200. El número entero distinto sirve para resolver conflictos a favor del proponente con el componente más distinguido. Además del componente monótono y el componente distinguido, el número redondo contiene un componente aleatorio. Un número redondo de esta manera asegura que uno de una pluralidad de proponentes competidores no se adelantará al usar el mismo número redondo para propuestas competidoras (es decir, a través del componente distintivo del número redondo) y asegura que la resolución del conflicto no favorezca o desaprobe perpetuamente a ninguno de los proponentes en particular (es decir, a través del componente aleatorio del número redondo).

Un mecanismo para comparar dos números redondos funciona de la siguiente manera. El número redondo con el componente monótono más grande es más grande que el otro. Si los componentes monótonos de los dos números redondos son iguales, el número redondo con el componente aleatorio mayor es mayor que el otro. Si las dos comparaciones anteriores no distinguen los números redondos, el número redondo con el componente distinguido más grande es más grande que el otro. Si las tres comparaciones anteriores no distinguen los números redondos, los números redondos son iguales.

Reclamación de almacenamiento persistente de manera eficiente

Con referencia a las figuras 3 y 4, los registros en el almacén de propuestas persistentes 230 de una máquina de

estado replicada 200 se organizan en grupos. Cada grupo almacena registros de propuestas que son propuestos con números de secuencia locales contiguos 330. Por ejemplo, los registros con los números de secuencia locales del #1 al #10000 pueden pertenecer al grupo-1, los registros con los números de secuencia locales #10001 a #20000 pueden pertenecer al grupo-2, y así sucesivamente.

5 Refiriéndose a grupos de propuestas persistentes, cada grupo puede almacenarse de tal manera que los recursos de almacenamiento utilizados por todo el grupo puedan recuperarse de manera eficiente. Por ejemplo, en un sistema de almacenamiento basado en archivos, cada grupo usa su propio archivo o conjunto de archivos.

10 Todavía con referencia a grupos de propuestas persistentes, el recuperador de almacenamiento 290 rastrea las solicitudes para eliminar registros individuales, pero no elimina registros individuales en el momento de las solicitudes. Cuando las solicitudes acumuladas para eliminar registros individuales incluyen todos los registros en un grupo, el recuperador de almacenamiento 290 reclama eficientemente los recursos de almacenamiento utilizados por el grupo. Por ejemplo, en un sistema de almacenamiento basado en archivos, el archivo o conjunto de archivos
15 utilizados por el grupo puede ser eliminado.

Los registros en el almacén de acuerdos 245 de la máquina de estado replicada 200 están organizados en grupos. Cada grupo almacena registros de instancias de protocolo de acuerdo con números de instancia de acuerdo contiguos 150. Por ejemplo, los registros con los números de instancia de acuerdo #1 a #10000 pueden pertenecer al grupo-1, los registros con los números de instancia de acuerdo #10001 a #20000 pueden pertenecer al grupo-2, y así sucesivamente.

20 Refiriéndose a grupos de instancias de protocolo de acuerdo, cada grupo puede almacenarse de tal manera que los recursos de almacenamiento utilizados por todo el grupo puedan recuperarse de manera eficiente. Por ejemplo, en un sistema de almacenamiento basado en archivos, cada grupo usa su propio archivo o conjunto de archivos.

Todavía con referencia a grupos de instancias de protocolo de acuerdo, el recuperador de almacenamiento 290 rastrea las solicitudes para eliminar registros individuales, pero no elimina registros individuales en el momento de las solicitudes. Cuando las solicitudes acumuladas para eliminar registros individuales incluyen todos los registros en un grupo, el recuperador de almacenamiento 290 reclama eficientemente los recursos de almacenamiento utilizados por el grupo. Por ejemplo, en un sistema de almacenamiento basado en archivos, el archivo o conjunto de archivos
30 utilizados por el grupo puede ser eliminado.

Gestión de propuestas pequeñas de manera eficiente

35 Con referencia a las figuras 3 y 4, una máquina de estado replicada de acuerdo con una realización (por ejemplo, la máquina de estado replicada 200) agrupa la transmisión de las propuestas que se han propuesto a la máquina de estado replicada 200 desde uno de los nodos de aplicación locales 205 a los receptores de los nodos de aplicación locales 205. Tal práctica permite que una máquina de estado replicada de acuerdo con una realización utilice eficientemente un protocolo de comunicación basado en paquetes en una situación en la que el tamaño de las propuestas de la máquina de estado replicada es pequeño en relación con el tamaño de un paquete de datos en el paquete subyacente basado en el protocolo de comunicación utilizado por la máquina de estado replicada.

45 En una realización, tal lote de propuestas puede ser tratado como una propuesta única por el protocolo del acuerdo. De esta manera, en cada nodo local 205, mientras que una máquina de estado replicada respectiva 200 determina el número de acuerdo 350 de un primer lote de propuestas que se han propuesto, Las propuestas que se han propuesto en el respectivo nodo de aplicación local 205 pueden acumularse en un segundo lote de propuestas. Cuando se determina el acuerdo número 150 del primer lote, la máquina de estado replicada 200 inicia la determinación del número de instancia de acuerdo 350 del segundo lote, y las propuestas que se han propuesto en ese nodo de aplicación local 205 se acumulan en un tercer lote, y así sucesivamente.

Gestión de grandes propuestas 110 de manera eficiente

55 Para reducir el ancho de banda de la red para grandes propuestas, una máquina de estado replicada de acuerdo con una realización permite que las propuestas se etiqueten con un ID de propuesta breve (por ejemplo, una identificación única global de 16 bytes) y/o las propuestas se pueden codificar en un formato denominado propuesta basada en archivos. Por el contrario, las grandes propuestas presentan un problema para las máquinas de estado replicadas convencionales, ya que tales propuestas grandes se envían esencialmente varias veces a través de una red, según el protocolo de acuerdo de una máquina de estado replicada convencional. Tal transmisión múltiple
60 puede no ser preferible porque el tamaño de las grandes propuestas puede ser de varios megabytes o incluso gigabytes.

Al transmitir grandes propuestas, una realización solo transmite identificadores de propuesta cortos una vez que la propuesta real se ha transmitido con éxito a un punto final de la red. Las propuestas basadas en archivos esencialmente llevan un puntero de archivo en memoria, mientras que el contenido de la propuesta real puede mantenerse en el disco en un archivo. Al transportar dicha propuesta basada en archivos en la red, una máquina de

estado replicada de acuerdo con una realización usa un protocolo eficiente de transmisión de archivos tolerante a fallos. Tal transporte puede ser gestionado por la capa DFTP 250 de una máquina de estado replicada 200 (figura 3). La capa DFTP 250 rastrea la propuesta basada en archivos de pares y un punto final de red. Asegura que una propuesta basada en archivos solo se transmita una vez a un punto final de la red. En caso de fallos que conduzcan a transferencias parciales, la propuesta basada en archivos se puede recuperar de cualquier punto final disponible que tenga la parte requerida del archivo.

En una realización, la implementación de DFTP utiliza archivos de envío nativos o archivos mapeados en memoria para una transferencia de archivos eficiente si el sistema operativo admite estas características. Si el remitente original no es accesible por un nodo que requiera un archivo, ese nodo localizará un remitente alternativo, un nodo diferente en el sistema que tiene el archivo. Cuando se opera sobre el protocolo TCP, DFTP utiliza múltiples conexiones TCP para aprovechar al máximo las conexiones de alto ancho de banda que también están sujetas a una alta latencia. Además, para aprovechar al máximo las conexiones de alto ancho de banda que también están sujetas a una alta latencia, un tamaño de ventana del protocolo TCP se puede ajustar de manera adecuada y/o deseable.

Pasando ahora a una discusión sobre la replicación escalable y activa de los depósitos de información, en una realización, la implementación de dicha replicación de acuerdo con una realización utiliza la máquina de estado replicada mencionada anteriormente. De manera más específica, proporcionar dicha replicación de acuerdo con una realización impacta ventajosamente en la escalabilidad, fiabilidad, disponibilidad y tolerancia a fallos de una máquina de estado replicada. En consecuencia, la implementación de una máquina de estado replicada de acuerdo con una realización impacta ventajosamente dicha replicación en una arquitectura de sistema informático distribuido. Al implementar la replicación de un depósito de información de acuerdo con una realización, se cumplirán todos o una parte de los siguientes objetos: permitir la replicación de un depósito CVS, una base de datos o cualquier depósito de información en general; permitiendo el uso concurrente, incluyendo modificación, de todas las réplicas de un depósito de información; preservando la consistencia de las réplicas a pesar de fallos esencialmente arbitrarios o fallos parciales de las redes de ordenadores utilizadas en la infraestructura de replicación: preservar la consistencia de las réplicas a pesar de fallos esencialmente arbitrarios o fallas parciales de los ordenadores o recursos informáticos asociados con las réplicas; asegurando la disponibilidad continua del depósito de información a pesar de fallas significativas de la naturaleza descrita anteriormente; permitiendo la distribución geográfica de las réplicas de modo que no haya restricciones sobre la distancia (por ejemplo, en diferentes continentes) o qué tan cerca (por ejemplo, en el mismo centro de datos, o incluso en el mismo estante) las réplicas están entre sí: permitiendo que todas las réplicas del depósito de información en conjunto manejen una carga mayor de la que puede manejar una instancia del depósito: preservar una equivalencia de copia de las réplicas: permitiendo la replicación del depósito de información sin introducir un solo punto de fallo en el sistema; permitiendo la replicación de un depósito de información sin modificaciones a las implementaciones del depósito de información: permitiendo la replicación de un depósito de información sin modificaciones a las implementaciones de los clientes del depósito de información: ofreciendo a los clientes de un depósito CVS tiempos de respuesta de un CVS local colocado depósito mediante quórum giratorio de réplica; reduciendo la comunicación de red entre los clientes del depósito CVS y el depósito CVS remoto en un factor de aproximadamente 3 en una red de área amplia (por ejemplo, aproximadamente 4,5 viajes de ida y vuelta a aproximadamente 1,5 viajes de ida y vuelta): permitiendo la recuperación remota de réplicas fallidas de manera automatizada sin requerir la intervención del administrador: y asegurando la limpieza distribuida del estado de todas las réplicas de manera automatizada sin requerir la intervención del administrador.

Con referencia a la figura 7, se muestra una realización de un replicador de acuerdo con una realización, que se denomina en el presente documento como el replicador 600. El replicador 600 consta de una pluralidad de módulos funcionales, incluyendo una interfaz de cliente replicador 610, un precalificador 620, una máquina de estado replicada 630, un programador 640, una interfaz de depósito de replicador 650, un administrador de resultados 660 y una consola de administrador 670. La interfaz del cliente replicador 610, el precalificador 620, la máquina de estado replicada 630, el programador 640, la interfaz del depósito del replicador 650, el administrador de resultados 660 y la consola de administrador 670 están interconectados cada uno al menos a una parte de los otros módulos para permitir la interacción entre los mismos. La máquina de estado replicada 200, cuya funcionalidad se discutió en referencia a las figuras 2-6, es un ejemplo de la máquina de estado replicada 630 del replicador 600. Por lo tanto, la máquina de estado replicada 630 es fiable, disponible, escalable y tolerante a fallos.

La figura 8 muestra una realización del despliegue del replicador 600 dentro de una arquitectura de sistema informático de múltiples sitios de acuerdo con una realización. La arquitectura del sistema informático de múltiples sitios puede incluir una pluralidad de sistemas de aplicaciones distribuidas 601. Cada sistema de aplicaciones distribuidas 601 puede incluir una pluralidad de clientes 680, un replicador 600, una interfaz de cliente de depósito 690, un depósito 695 (es decir, un depósito de información) y una red 699. La red 699, que generalmente no es necesariamente un componente de una pluralidad de sistemas de aplicaciones distribuidas 601, puede conectarse entre los clientes 680 de cada sistema de aplicaciones distribuidas 601 y el respectivo replicador 600 y entre la interfaz de cliente del depósito 690 de cada sistema de aplicaciones distribuidas 601 y el respectivo replicador 600, interconectando así los clientes 680, el replicador 600 y el depósito 695 de cada sistema de aplicaciones distribuidas 601 para permitir la interacción de dichos componentes de cada sistema de aplicaciones distribuidas 601. La red también puede estar conectada entre el replicador 600 de todo el sistema de aplicaciones distribuidas 601,

permitiendo así la interacción entre todo el sistema de aplicaciones distribuidas 601. Las redes 699 pueden aislarse entre sí, pero no necesitan serlo. Por ejemplo, la misma red puede cumplir los tres roles descritos anteriormente.

5 Como se muestra en la figura 8, tres clientes 680 están "cerca" de cada uno de los depósitos 695 (es decir, un elemento del sistema de los sistemas de aplicaciones distribuidas 601 que comprende un depósito respectivo 695). Por cerca, se quiere decir que un cliente particular 680 cerca de un depósito particular 695 preferiría acceder a ese depósito particular 695. Alternativamente, ese cliente particular de los clientes 680 podría acceder potencialmente al depósito 695 de cualquiera de los sistemas de aplicaciones distribuidas 601.

10 Los operadores de un sistema informático distribuido de acuerdo con una realización incluyen los usuarios del cliente 680 y el administrador o administradores de los sistemas de aplicaciones distribuidas 601. Los usuarios del cliente 680 siguen las instrucciones del manual del usuario de su cliente. Un usuario podría permanecer ajeno al hecho de que está utilizando un replicador de acuerdo con una realización, muchos de los aspectos ventajosos de las realizaciones pueden ser transparentes para el usuario. Un administrador, además de las tareas estándar de administrar el propio depósito 695, configurarán las redes en consecuencia, según sea necesario y si es necesario para la operación.

20 Las máquinas de estado replicadas 630 de cada sistema de aplicaciones distribuidas 601 se comunican entre sí a través de la red 699. Cada interfaz de depósito de replicador 650 interactúa a través de la red 699 con el depósito 695 del respectivo sistema de aplicaciones distribuidas 601. El cliente 680 interactúa a través de la red 699 con la interfaz del cliente replicador 610. Opcionalmente, un producto tal como, por ejemplo, Cisco Systems Director se puede utilizar para permitir que un cliente particular 680 de uno particular de los sistemas de aplicaciones distribuidas 601 falle en cualquiera de los otros sistemas de aplicaciones distribuidas 601, si el sistema de aplicaciones distribuidas 601 que comprende el cliente 680 puede no estar disponible en un momento particular para proporcionar una funcionalidad requerida.

30 Con referencia ahora a las figuras 7 y 8, la interfaz del cliente replicador 610 puede ser responsable de la interfaz con un cliente particular 680 (es decir, el cliente particular 680) asociado con un depósito dirigido 695. La interfaz del cliente replicador 610 reconstruye los comandos emitidos por el cliente particular 680 a través de la red 699 y entrega los comandos al precalificador 620. El precalificador 620 permite una operación eficiente del replicador 600, pero puede no ser necesario para la operación útil y ventajosa del replicador 600.

35 Para cada comando, el precalificador 620 puede determinar opcionalmente si el comando está condenado a fallar, y si es así, determinar un mensaje de error apropiado o un estado de error que se devolverá al cliente particular 680. Si es así, ese mensaje de error o estado de error puede devolverse a la interfaz del cliente del replicador 610 y la interfaz del cliente del replicador 610 entrega ese mensaje de error o estado de error al cliente particular. 680. Posteriormente, el replicador 600 ya no puede procesar el comando.

40 Para cada comando, el precalificador 620 puede determinar opcionalmente si el comando puede omitir la máquina de estado replicada 630 o tanto la máquina de estado replicada 630 como el programador 640. Si el precalificador 620 no determinó que la máquina de estado replicada 630 podría ser ignorada, el comando se puede entregar a la máquina de estado replicada 630. La máquina de estado replicada 630 recopila todos los comandos enviados a la misma y sus máquinas de estado replicadas pares 630 en el replicador asociado 600 del sistema de aplicaciones distribuidas 601. Se puede asegurar que esta secuencia de operaciones sea idéntica en todos los sistemas de aplicaciones distribuidas 601. En cada uno de los sistemas de aplicaciones distribuidas 601, la respectiva máquina de estado replicada 630 entrega los comandos recopilados como se indica anteriormente, en secuencia, al respectivo programador 640.

50 El programador 640 realiza un análisis de dependencia de los comandos que se le entregan y determina el orden parcial más débil de los comandos que aún garantizarían la serialización de una copia. Dicho análisis de dependencia y serialización de una copia se divulgan en la referencia de la técnica anterior de Wesley Addison titulada "Concurrent Control & Recovery in Database Systems" y publicado en un libro de referencia de P. Berstein et. al. El programador 640 luego entrega los comandos a la interfaz del depósito del replicador 650, concurrentemente cuando lo permita el orden parcial construido, secuencialmente de lo contrario.

55 La interfaz del depósito del replicador 650 entrega los comandos al depósito 695. En respuesta, se produce uno de los tres resultados. Posteriormente, la interfaz del depósito de replicador 650 entrega el resultado resultante al administrador de resultados 660.

60 Uno de los primeros resultados puede incluir el depósito 695 que devuelve una respuesta al comando. Esta respuesta contiene un resultado, un estado o ambos, indicando que nada salió mal durante la ejecución del comando. Si el comando se originó localmente, el administrador de resultados 660 entrega la respuesta a la interfaz del cliente replicador 610, que a su vez entrega la respuesta al cliente 680. Si el comando se originó en un replicador de un sistema de aplicaciones distribuidas diferente 601, la respuesta se descarta preferiblemente.

65 Un segundo de los resultados puede incluir que el depósito 695 responda con un estado de error. El administrador

de resultados 660 determina si el estado de error indica un error determinista en el depósito 695 (es decir, si ocurriría el mismo error o un error comparable en cada uno de los otros sistemas de aplicaciones distribuidas 601). Si la determinación del error puede ser ambigua, el administrador de resultados 660 intenta comparar el error con el resultado en otros sistemas de aplicaciones distribuidas 601. Si esto no resuelve la ambigüedad, o si el error puede ser inequívocamente no determinista, el administrador de resultados 660 suspenderá la operación del replicador 600 e informará al operador a través de la consola de administrador 670 (es decir, mediante la emisión de una notificación a través de la consola administrativa 670).

En el caso de que el replicador sea un replicador CVS, como se describe a continuación en referencia a la funcionalidad específica de CVS, el administrador de resultados puede usar una lista de patrones de error para marcar el error determinista. El administrador de resultados 660 usa estos patrones para hacer una coincidencia de expresión regular en la secuencia de respuesta.

Un tercer resultado puede incluir el depósito 695 colgado (es decir, no regresa de la ejecución del comando). En una realización, este resultado puede tratarse exactamente como un error no determinista como se discutió en referencia al segundo de los resultados.

De acuerdo con una realización, cada replicador 600 se puede configurar alternativamente. En una realización alternativa, el replicador 600 puede estar integrado y controlado directamente por el cliente 680 del depósito 695. En otra realización alternativa, el replicador 600 puede estar integrado en la interfaz del cliente 690 al depósito 695. En otra realización alternativa, el secuenciador global del replicador (por ejemplo, el secuenciador global 280 que se muestra en la máquina de estado replicada 200 en la figura 3) puede basarse en otras tecnologías, con los compromisos correspondientes de robustez y calidad de servicio. Uno de los varios ejemplos posibles de dicha tecnología es la comunicación grupal. En otra realización alternativa, el replicador 600 maneja más de un depósito 695, con el compromiso correspondiente de robustez y calidad de servicio. En otra realización alternativa, los módulos del replicador 600 se fusionan en módulos más de grano grueso, dividido en módulos más finos, o ambos. En otra realización alternativa, como salvaguardia redundante contra la desviación de la posibilidad de una copia en serie, las respuestas de todos los sistemas de aplicaciones distribuidas 601 se comparan para garantizar que la información contenida en los depósitos 695 de cada sistema de aplicaciones distribuidas 601 permanezca consistente con respecto al otro sistema de aplicaciones distribuidas 601.

En referencia a las figuras 7 y 8. cada uno de los depósitos 695 discutidos anteriormente puede ser un depósito del Sistema de Versiones Simultáneas (CVS) y los clientes 680 pueden ser, en consecuencia, clientes CVS. Donde los depósitos 695 son depósitos CVS y los clientes 680 son clientes CVS, las interfaces asociadas con los depósitos 695 y los clientes 680 son interfaces específicas de CVS (por ejemplo, una interfaz de cliente CVS de replicador, una interfaz de depósito CVS de replicador y una interfaz de cliente CVS de depósito). Además, de acuerdo con una realización, el replicador 600 se puede modificar para incluir una funcionalidad que esté específica y especialmente configurada para su uso con un depósito CVS.

La interfaz del cliente replicador 610 descrita en el presente documento puede configurarse específicamente para interactuar con un cliente CVS de un depósito CVS objetivo. Con este fin, la interfaz del cliente del replicador 610 almacena bytes entrantes del cliente CVS en una memoria intermedia de archivo mapeado en memoria. La interfaz del cliente del replicador 610 detecta el final del comando CVS cuando ve una cadena de comando válida en la secuencia de bytes entrante. Una lista no limitativa de tales cadenas de comando válidas puede incluir, pero no limitado a, "Raíz", "Respuestas válidas", "solicitudes válidas", "Depósito", "Directorio", "Max-dotdot", "Directorio estático", "Pegajoso", "Entrada", "Kopt", "Hora de entrada", "Modificado", "Está modificado", "Uso sin cambios", "Sin cambios", "Notificar", "Cuestionable", "Argumento", "Argumentox", "Opción_global", "Corriente_Gzip", "Envíame-opciones-envoltura-rc-s", "Conjunto", "expandir-módulos", "ci", "co", "actualizar", "dif", "inicio de sesión", "inicio de sesión r", "lista", "lista r", "silenciar-lista-global", "Es", "añadir", "retirar", "parches de actualización", "contenido del archivo gzip", "estado", "rdif", "etiqueta", "etiqueta r", "importar", "administrar", "exportar", "historia", "lanzamiento", "ver en", "vigilancia", "ver adición", "ver retirada", "vigilantes", "editores", "iniciar", "anotar", "anotar r", "noop" y "versión".

La interfaz del cliente del replicador 610 intenta clasificar el comando CVS entrante como un comando de lectura o un comando de escritura. Una lista no limitativa de cadenas de comandos de escritura válidas puede incluir, pero no limitado a, "ci", "etiqueta", "etiqueta r", "administrar", "importar", "añadir", "retirar". "ver en", "vigilancia" e "iniciar". Cualquier comando dentro de la lista de cadenas de comando válidas que no pertenece a la lista de cadenas de comando de escritura válidas se considera aquí como una cadena de comando de lectura con respecto a la lista de cadenas de comando válidas.

Los comandos de lectura se entregan directamente a la interfaz del depósito del replicador CVS para su ejecución por el depósito CVS de destino. Los comandos de escritura CVS se entregan opcionalmente al módulo de precalificador 20.

Para cada comando de escritura CVS, el módulo de precalificador 20 puede determinar opcionalmente si el

comando CVS está condenado a fallar, y si es así, determinar un mensaje de error apropiado o un estado de error que se devolverá al cliente CVS. La detección de fallos puede basarse en hacer coincidir el resultado o el flujo de bytes de estado devuelto por el depósito CVS con patrones de error conocidos. Se incluyen ejemplos de patrones de error del sistema conocidos, pero no se limitan a, no puede crear un enlace simbólico de *.a.*; no se puede iniciar el servidor a través de rsh; no puede fstat.*; no se pudo crear el archivo temporal; no se puede abrir el archivo dbm.* para la creación; no puede escribir a.*; no se puede registrar el archivo de historial; no se puede abrir el archivo de historial.*; no se puede abrir *.*; no se pudo registrar el archivo RCS.* para mapeo; no se puede abrir el archivo.* para comparar: memoria virtual agotada; no puede ftello en el archivo RCS.*; no puedo leer *.*; no se puede obtener la lista de grupos auxiliares: no se puede sincronizar el archivo .* después de la copia: no se puede stat *.*; no se puede abrir el directorio actual: no se pueden hacer estadísticas del directorio .*; no puede escribir *.*; no se puede leer el enlace *.*; no puede cerrar la tubería; no puede cambiar al directorio .*; no se puede crear un archivo temporal; no se pudo obtener información del archivo para *; no se pudo abrir el archivo de salida dif *.*; no se puede crear *.*; no se puede obtener el directorio de trabajo: no se puede lstat *.*; horquilla para diff falló en *.*; no se pudo obtener información para no se puede cambiar el modo para *.*; no puede ftello para *.*; Error de verificación del mensaje: no se puede registrar el archivo temporal *.* sin memoria; no se puede crear el directorio .* en *.*; iniciar sesión: Error al leer la contraseña; error al leer el archivo de historial; no se pudo obtener el directorio de trabajo; no se puede establecer el indicador close-on-exec en \d+; error al escribir en el archivo de bloqueo *.*; no se puede escribir en el archivo de historial: *.*; no se puede cambiar el nombre del archivo .* a *.*; no puede cambiar al directorio .*; no se puede obtener información del archivo para *; no se puede crear .* para copiar; no se puede escribir el archivo temporal *.*; no se puede abrir *.*; lectura de control de flujo fallida; escribiendo al servidor; no se puede cerrar *.*; no se pudo abrir el archivo de bloqueo, no se puede abrir \d+ para leer; no se puede cerrar el archivo temporal *.*; no cambia el directorio al directorio de pago solicitado *.*; no se puede crear el directorio *.*; valor de umask no válido en; no se pudo abrir *.* para leer; incapaz de obtener el número de grupos auxiliares; no se puede abrir *.* para escribir; no se puede chdir a *.*; la bifurcación falló al diferir *.*; no se podría no abrir *.*; no se puede abrir \d+ para escribir; escribir a *.* falló; no se puede crear un archivo temporal *.*; no se puede leer *.*; no se puede escribir el archivo *.* para copiar; no se puede abrir *.* para copiar; no puede duplicar tubería; no puede entrar en *.*; no se puede abrir *.* para escribir; no se puede bifurcar; error al escribir en el servidor: no se puede registrar *.* --bifurcación fallida; no se puede leer el archivo .* para comparar: no se puede vincular .* a *.*; error al cerrar *.*; no puede duplicar la conexión de red; lectura de datos fallida; no se puede leer *.*; no se puede eliminar *.*; no pudo chdir a *.*; no se puede abrir el archivo temporal *.*; no se puede stat *.*; no se puede abrir el directorio *.*; fwrite falló; no se puede crear el archivo temporal *.*; no se puede registrar el archivo temporal; no se puede stat *.*; no se puede leer *.*; error definiendo *.*; no se puede crear un archivo especial *.*; no se puede cerrar el archivo de historial: *.*; no se puede asignar memoria al archivo RCS *; no se puede crear el directorio *.*; no se puede leer el archivo .* para copiar; no se puede crear la tubería: no se puede abrir el archivo temporal *.*; no se puede eliminar el archivo *.*; no se puede abrir; no puede buscar el final del archivo de historial: *.*; no puede chdir a *.*; lectura de longitud fallida; no se puede ejecutar *.*: no se puede fdopen .* y no se puede encontrar el tamaño del archivo temporal. Se incluyen ejemplos de patrones de error no relacionados con el sistema conocidos, pero no se limitan a, error interno; no existe tal depósito; no se pudo encontrar la versión deseada; nombre de pila fallido; advertencia: ferror establecido al reescribir el archivo RCS: error interno: a islink no le gusta readlink; acceso denegado: no se pueden comparar archivos del dispositivo en este sistema: error interno del servidor: caso no manejado en servidor_actualizado; recibida señal *.*: error interno: no hay información de revisión para; error de protocolo: Modo duplicado; error interno del servidor: no hay modo en servidor_actualizado; caché rcsbuf abierto: error interno: Error fatal, abortar; error fatal: salir: *.*; EOF inesperado: *.*; número de revisión confuso; archivo rcs inválido: EOF en clave en archivo RCS; Los archivos RCS en CVS siempre terminan en v; perdió la información del enlace duro para; no se puede leer *.*; fin del archivo: resbuf abierto: error interno; sin memoria; no puede asignar infopath; susurros finales de *.* inesperados; error interno: mala fecha *.*; la autenticación kerberos falló: *.*; *, delta *.*; EOF inesperado; EOF inesperado leyendo el archivo RCS *.*; ERROR: fuera de aborto espacial; control de flujo EOF; no se puede ver el archivo RCS *.*; fallo de suma de control en *.*; Error interno de CVS: estado desconocido \d+; error interno: argumento incorrecto para ejecutar impresión; no se pueden copiar archivos del dispositivo en este sistema; final inesperado de la lectura del archivo *.*; sin memoria; error interno: no se analizó el archivo RCS; error interno: EOF demasiado temprano en RCS_copydeltas; error interno: prueba de soporte para respuesta desconocida\?; EOF en valor en el archivo RCS *.*; PANIC* faltan archivos de administración \!: final prematuro de la lectura del archivo *.*; EOF mientras busca valor en el archivo RCS no puede continuar; bloqueo de lectura fallido abandonando; lectura inesperada de EOF *.*; no puede resucitar *.*; archivo RCS eliminado por una segunda parte; su nombre de usuario aparente *.* es desconocido para este sistema; corrupción de la base de datos de atributos de archivo: falta la pestaña en *.*; no se puede importar *.*; incapaz de importar archivos de dispositivo en este sistema; no se puede importar *.*; tipo desconocido de archivo especial: no se puede importar *.*: archivo especial de tipo desconocido; ERROR: no se puede mkdir *.* --no añadido; no se puede crear bloqueo de escritura en el depósito *.*: no se puede crear *.*: no se pueden crear archivos especiales en este sistema; no se puede preservar *.*: no se pueden guardar archivos del dispositivo en este sistema; error al analizar el archivo del depósito *.* el archivo puede estar dañado y se desconoce el estado del archivo \d+ para el archivo *.*.

Como se discutió anteriormente en referencia a las figuras 7 y 8, para cada comando, el módulo de precalificador 620 puede determinar que el comando está condenado a fallar y puede omitir tanto la máquina de estado replicada 630 como el programador 640. En el caso de la funcionalidad específica de CVS, si el módulo de precalificador 620 no determinara que la máquina de estado replicada 630 podría ser ignorada, el comando puede convertirse en un

comando de propuesta CVS. El comando de propuesta CVS contiene la matriz de bytes del comando CVS real, así como un conjunto de bloqueos que describe los bloqueos de escritura que este comando CVS haría que el depósito CVS obtuviera si lo ejecutara directamente. Como se describe a continuación, el programador 640 utiliza este conjunto de bloqueo.

5 El comando de propuesta CVS se puede entregar a la máquina de estado replicada 630. La máquina de estado replicada 630 recopila todos los comandos enviados a la misma y sus máquinas de estado replicadas pares 630 en cada uno de los otros replicadores, en una secuencia. Se garantiza que esta secuencia sea idéntica en todas las réplicas. En cada uno de los sistemas de aplicaciones distribuidas 601, la máquina de estado replicada 630 entrega los comandos recopilados como se indica anteriormente, en secuencia, al programador 640.

15 El programador 640 realiza un análisis de dependencia de los comandos que se le entregan y determina el orden parcial más débil de los comandos que aún garantizarían la serialización de una copia. El programador 640 entrega los comandos a la interfaz del depósito del replicador CVS, concurrentemente cuando lo permita el orden parcial construido, secuencialmente de lo contrario.

20 De acuerdo con una realización, el análisis de dependencia puede basarse en pruebas de conflictos de bloqueo. Cada comando de propuesta CVS enviado al programador contiene un conjunto de bloqueo. El programador garantiza que se entregue un comando a la interfaz del depósito CVS si y solo si ningún otro conjunto de bloqueo de comando entra en conflicto con su conjunto de bloqueo. Si se detecta un conflicto, el comando espera en la cola para ser programado en un último punto cuando se pueden obtener todos los bloqueos del conjunto de bloqueo sin conflictos.

25 Como se divulgó anteriormente, la implementación de una arquitectura de sistema informático de múltiples sitios impacta ventajosamente en la escalabilidad, fiabilidad, disponibilidad y tolerancia a fallos de tales máquinas de estado replicadas. El escalado eficiente requiere procesos eficientes para agregar nuevos nodos de aplicaciones distribuidas (o simplemente, nodos) al sistema. Nodos recién agregados, sin embargo, se les debe dar una cierta cantidad de información para que puedan participar en el sistema informático distribuido. Por ejemplo, un nuevo nodo debe recibir las credenciales necesarias para unirse al proyecto colaborativo y debe informarse sobre las ubicaciones y los nodos existentes que deben ser visibles para el mismo y con quién el nodo recién invitado puede intercambiar mensajes e interactuar. De acuerdo con una realización, esto se logra mediante un modelo de mensajería y métodos de inducción de nodos y dispositivos y sistemas correspondientes que son efectivos para permitir que un nodo inductor traiga un nodo inducido al sistema informático distribuido y permita que el nodo inducido realice un trabajo útil.

35 Modelo de mensajería

40 En el presente documento, debe entenderse que el término "inductor" o "nodo inductor" se refiere a un nodo que al menos inicia la inducción de otro nodo, el "nodo inducido" en el sistema informático distribuido. De acuerdo con una realización, se supone que los nodos inductor e inducido se comunican entre sí mediante el envío de mensajes mediante un modelo asíncrono no bizantino donde:

- 45 - Cualquiera de los procesos puede operar a una velocidad arbitraria, puede fallar deteniéndose y puede reiniciarse:
- Dado que un proceso puede fallar en cualquier momento, cierta información debe ser recordada (es decir, ser persistente) entre reinicios: y
- 50 - Los mensajes pueden demorar mucho tiempo en ser entregados, pueden duplicarse y perderse, pero los mensajes no están dañados (ya que un mensaje dañado se trata igual que un mensaje no entregado, ya que el receptor lo descartará).

55 La figura 9 es un diagrama que muestra aspectos de los dispositivos, métodos y sistemas que permiten una inducción segura y autorizada de un nodo en un grupo de nodos de acuerdo con una realización. Como se muestra allí y de acuerdo con una realización, puede comprender un método para inducir un nodo en un sistema informático distribuido, y los sistemas y dispositivos actuales pueden configurarse para ejecutarse, varias fases tales como, por ejemplo, una fase de preautorización. Fase de inicio del reclutado, un despliegue de una fase de membresía de inicio y un conocimiento del nodo y la ubicación de la persona ingresada. Además, se pueden realizar una pluralidad de tareas posteriores a la inducción. Cada una de estas fases se describe en detalle a continuación.

60 A. Fase de preautorización

65 De acuerdo con una realización, la fase de preautorización puede llevarse a cabo antes de que se inicie el nodo 206 del inducido y puede proporcionar la oportunidad para que un administrador 202 cree una tarea de inducción que puede comprender información para ser utilizada en el proceso de inducción y permitir la preconfiguración del proceso de inducción para que pueda, de acuerdo con una realización, proceder sin ninguna interacción humana.

A.0 Creación de una nueva tarea de inducción

5 Antes de que se inicie el nodo inducido 206, se puede crear una tarea de inducción en el nodo inductor 204 que contiene la información requerida para un proceso de inducción exitoso y completo. El uso de una tarea persistente permite que la información requerida en el proceso de inducción se almacene en el mismo lugar, para que esta información y el estado del proceso de inducción persistan en los reinicios del nodo inductor 204 y para que la misma tarea de inducción se copie (clone) y se reutilice en otras inducciones.

10 De acuerdo con una realización, una tarea de inducción puede configurarse para comprender tres elementos: un ticket de inducción: el conjunto de nodos de los cuales el nodo 206 inducido debe ser consciente: y un conjunto de tareas posteriores a la inducción. Debe entenderse que se pueden agregar o sustituir otros elementos por estos tres elementos.

15 A.1 El ticket de inducción

Se puede generar una inducción, por ejemplo, por un administrador y enviado al nodo inducido 206, como se muestra en B21 en la figura 9. Este ticket de inducción proporciona un mecanismo para que el administrador 202 empaquete los detalles de contacto del nodo inductor 204, especifique (y controle) los detalles del nuevo nodo y también para especificar algunos otros parámetros de configuración de la plataforma o aplicación para el nuevo nodo, por ejemplo. De acuerdo con una realización, el ticket de inducción puede comprender:

- la identidad de la tarea de inducción;
- 25 - el nodo y la identidad de ubicación del nodo inducido 206;
- la identidad de la ubicación, nombre del huésped y puerto del nodo inductor 204 (la información básica necesaria para que el nodo inducido 206 contacte con el nodo inductor 204); y/o
- 30 - otra información arbitraria de configuración de plataforma/aplicación.

El ticket de inducción puede comprender otra información que logre el mismo resultado o un resultado funcional similar al permitir que el nodo inductor 206 vea, sea visible y se comunique con otros nodos seleccionados en el sistema informático distribuido. El ticket de inducción se puede configurar, por ejemplo, como un archivo. Para mejorar la seguridad, este ticket de inducción, por lo tanto, se puede firmar con código utilizando la clave privada del nodo inductor en un sistema PKI. A su vez, el nodo inducido 206 puede configurarse para validar la autenticidad de los detalles contenidos en el boleto de inducción usando la clave pública del nodo inductor 204. Se pueden utilizar otros métodos de autenticación y definición de autoridad con buenos resultados, ya que las implementaciones descritas y mostradas en este documento no se limitan al modelo de seguridad PKI. El ticket de inducción puede, de acuerdo con una realización, enviarse posteriormente fuera de banda a un ingeniero 208 que realiza la instalación del nodo inducido 206. De acuerdo con una realización, el ticket de inducción puede permanecer con el nodo inductor 204 y puede ser "empujado" al nodo inducido 206 cuando se inicia el nodo inductor 206.

45 A.2 El conjunto de nodos que el nodo inducido debe tener en cuenta

De acuerdo con una realización, la tarea de inducción puede comprender detalles sobre los nodos existentes sobre los que se debe informar al nodo inducido 206 durante el proceso de inducción. El nodo inducido 206, se recuerda, se informa de los otros nodos dentro del sistema informático distribuido con el que el nodo inducido 206 está habilitado y/o permitido para comunicarse/trabajar. Dicha información, por lo tanto, puede especificarse ventajosamente antes de que se inicie el proceso de inducción si no hay interacción humana. La selección del nodo o nodos con los que se habilita o permite que el nodo inducido 206 se comunique puede llevarse a cabo utilizando una interfaz de usuario (UI) que permite al administrador 202 elegir un conjunto de nodos del conjunto completo o un subconjunto de nodos existentes que ya han sido incluidos en la red de nodos. Esta información puede almacenarse en la tarea de inducción, por lo que se puede acceder más tarde. La IU puede comprender, por ejemplo, un navegador o una aplicación de dispositivo móvil.

A. 3. Tareas posteriores a la inducción

60 De acuerdo con una realización, la tarea de inducción puede comprender detalles de una pluralidad de otras tareas, una o más de las cuales se pueden aplicar al nuevo nodo 206 inducido después de la inducción, como, por ejemplo, unirse a una membresía existente. Debe tenerse en cuenta que este conjunto de tareas puede estar vacío si el nodo inductor 206 no tiene que hacer nada después de la inducción. Una vez que la tarea de inducción se ha creado y persistido (por ejemplo, almacenada en una memoria no volátil), el nodo inducido 206 puede iniciarse.

65 B. Arranque de los inducidos

De acuerdo con una realización, un nodo inducido 206 puede iniciarse:

B.1. Sin el ticket de inducción presente en el nodo inducido:

5 De acuerdo con una realización, si el ticket de inducción no está presente en el nodo inducido 206, el nodo inducido 206 puede iniciarse o provocar que se inicie en una configuración básica y esperar (es decir, escuchar como se muestra en B22) para ser contactado por el nodo inductor 204 con detalles de la membresía de arranque de la cual el nodo inducido 206 se convertirá en miembro, como se describe a continuación.

10 B. 2. Con el ticket de inducción presente en el nodo inducido:

De acuerdo con una realización, si el ticket de inducción está presente en el nodo inducido 206 al inicio como se muestra en B23, el nodo inducido 206 puede configurarse para:

- 15 a) analizar (y opcionalmente validar, según corresponda) la información en el ticket de inducción.
- b) usar esta información para configurar la plataforma de la aplicación, y
- 20 c) usar esta información para crear y encender una baliza *SolicitudMembresíaArranque* como se muestra en B24 que puede configurarse para informar al nodo inductor 204 que, como se muestra en B25, el nodo inducido 206 está iniciando el proceso de inducción. De acuerdo con una realización, una baliza es un proceso configurado para transmitir repetidamente un mensaje a una lista predeterminada de destinatarios, eliminar los destinatarios objetivo de la lista predeterminada a la que se transmite el mensaje hasta que se reciba un acuse de recibo de respuesta de cada uno de los destinatarios objetivo. De acuerdo con una realización,
- 25 *SolicitudArranqueMembresía* puede configurarse para contener la identidad de la tarea de inducción y la identidad del nodo y la ubicación del inducido, nombre de hospedador y puerto.

De acuerdo con una realización, en respuesta al nodo inductor 204 que recibe la *SolicitudArranqueMembresía* del nodo inductor 206, el nodo inductor 204 puede enviar una *RespuestaArranqueMembresía* al nodo inductor 206 como se muestra en B26 para deshabilitar la baliza de solicitud, como se muestra en B27. El nodo inductor 204 puede buscar la tarea de inducción y verificar si el nodo y la identidad de ubicación coinciden con lo especificado en el ticket de inducción recibido anteriormente, como se muestra en B28. Si la verificación falla, es decir, el nodo y/o la identidad de ubicación no coinciden con los del ticket de inducción: el nodo inductor 204 puede transmitir un mensaje de *ArranqueMembresíaDenegada* al nodo inducido 206, como se muestra en B29.

30 Cuando el nodo inducido 206 recibe el mensaje *ArranqueMembresíaDenegada*, el nodo inducido 206 puede configurarse para enviar un mensaje de *AcusedereciboArranqueMembresía* en respuesta y finalizar, como se muestra en B30. Cuando el nodo inductor 204 recibe el mensaje *AcusedereciboArranqueMembresía* del nodo inductor 206 como se muestra en B31, el nodo inductor 204 puede deshabilitar la baliza

35 *ArranqueMembresíaDenegado*, como se muestra en B32.

C. Despliegue del arranque de membresía

45 De acuerdo con una realización, cuando el nodo inducido 206 se ha iniciado sin el ticket de inducción y el administrador 202 ha iniciado el proceso de inducción en el nodo inductor 204, o la búsqueda de la tarea de inducción ha sido exitosa, la creación y el despliegue de la membresía de arranque se puede llevar a cabo utilizando el siguiente proceso:

De acuerdo con una realización, el nodo inductor 204 puede, de acuerdo con una realización:

- 50 1. crear una membresía de arranque con:
- a. una identidad de membresía creada de manera determinista;
- 55 b. el nodo inductor 204 en el papel de Proponente de Acuerdo y Aceptador de Acuerdo;
- c. el nodo inducido en el papel de aprendiz.
2. desplegar la membresía como se muestra en B33;
- 60 3. crear una máquina de estado determinista que haga referencia a la membresía de arranque como se muestra en B34, y
4. enviar un mensaje *ArranqueMembresíaListo* al nodo inducido 206, como se muestra en B35.

De acuerdo con una realización, cuando el nodo inducido 206 recibe el mensaje *ArranqueMembresíaListo* como se muestra en B36 puede, de acuerdo con una realización:

65

1. crear una membresía de arranque con:

a) una identidad de membresía creada de manera determinista;

5 b) el nodo inductor 204 en el papel de Proponente de Acuerdo y Aceptador de Acuerdo;

c) el nodo inducido 206 en el papel de aprendiz.

2. implementar la membresía como se muestra en B37, y

10 3. crear una máquina de estado determinista que haga referencia a la membresía de arranque como se muestra en B38, y

4. enviar un mensaje *ArranqueMembresíaListo* al nodo inducido 206, como se muestra en B39.

15 De acuerdo con una realización, cuando el nodo inductor 204 recibe el mensaje *AcusedereciboArranqueMembresía*, debe deshabilitar la baliza *ArranqueMembresíaListo*, como se muestra en B40.

20 D. Conocimiento del nodo inducido y la ubicación

Después del despliegue de la membresía de arranque, el nodo inducido 206 puede ser informado de los nodos y ubicaciones de los que debería estar al tanto. Esto se puede lograr, de acuerdo con una realización, utilizando el siguiente proceso:

25 1. El nodo inductor 204 consulta la tarea de inducción para determinar qué ubicaciones y nodos se deben informar al nodo inducido 206;

2. La tarea de inducción que devuelve la lista de ubicaciones y nodos para este nodo inducido 206;

30 3. El nodo inductor 204 que propone a la máquina de estado determinista el conjunto de nodos y ubicaciones;

4. Cuando se forma un acuerdo como se muestra en B41, el nodo 206 del inducido que aprende sobre las ubicaciones y los nodos que necesita saber, como se muestra en B42.

35 5. Tras el aprendizaje del nodo inducido 206 de los nodos y ubicaciones, se completa el proceso de inducción.

E. Tareas posteriores a la inducción

40 Siguiendo el acuerdo de los nodos y ubicaciones, es decir, la finalización del proceso de inducción, - ahora debería ser posible ejecutar el conjunto de tareas especificado en la tarea de inducción. Estas tareas pueden comprender crear nuevas membresías que contengan el nodo recién inducido, unirse a membresías existentes (es decir, realizar un cambio de membresía para incluir el nodo recién incluido en una membresía existente) y realizar un despliegue y sincronización de una entidad replicada, por ejemplo.

45 La figura 10 ilustra un diagrama de bloques de un sistema informático 1000 sobre el cual se pueden implementar realizaciones. El sistema informático 1000 puede incluir un bus 1001 u otro mecanismo de comunicación para comunicar información, y uno o más procesadores 1002 acoplados con el bus 1001 para procesar información. El sistema informático 1000 puede comprender además una memoria de acceso aleatorio (RAM) u otro dispositivo de almacenamiento dinámico 1004 (denominado memoria principal), acoplado al bus 1001 para almacenar información e instrucciones para ser ejecutadas por los procesadores 1002. La memoria principal 1004 también se puede utilizar para almacenar variables temporales u otra información intermedia durante la ejecución de instrucciones mediante el procesador 1002. El sistema informático 1000 también puede incluir una memoria de solo lectura (ROM) y/u otro dispositivo de almacenamiento estático 1006 acoplado al bus 1001 para almacenar información estática e instrucciones para el procesador 1002. Un dispositivo de almacenamiento de datos 1007, tal como un disco magnético o una memoria Flash, por ejemplo, se puede acoplar al bus 1001 para almacenar información e instrucciones. El sistema informático 1000 también puede acoplarse a través del bus 1001 a un dispositivo de visualización 1010 para mostrar información a un usuario del ordenador. Un dispositivo de entrada alfanumérico 1022, que incluye teclas alfanuméricas y otras, puede estar acoplado al bus 1001 para comunicar información y selecciones de comandos al procesador o procesadores 1002. Otro tipo de dispositivo de entrada del usuario es el control del cursor 1023, como un ratón, una bola de seguimiento o teclas de dirección del cursor para comunicar información de dirección y selecciones de comandos al procesador 1002 y para controlar el movimiento del cursor en la pantalla 1021. El sistema informático 1000 puede estar acoplado a través de un dispositivo de comunicación (por ejemplo, módem, NIC) a una red 1026 y a uno o más nodos de un sistema informático distribuido.

65 Las realizaciones están relacionadas con el uso de un sistema informático y/o con una pluralidad de dichos sistemas informáticos para inducir nodos a un sistema informático distribuido. De acuerdo con una realización, los métodos y

sistemas descritos en el presente documento pueden ser proporcionados por uno o más sistemas informáticos 1000 en respuesta a los procesadores 1002 que ejecutan secuencias de instrucciones contenidas en la memoria 1004. Dichas instrucciones pueden leerse en la memoria 1004 desde otro medio legible por ordenador, tal como el dispositivo de almacenamiento de datos 1007. La ejecución de las secuencias de instrucciones contenidas en la memoria 1004 hace que los procesadores 1002 realicen las etapas y tengan la funcionalidad descrita en el presente documento. En realizaciones alternativas, circuitos cableados pueden usarse en lugar de o en combinación con instrucciones de software para implementar las realizaciones. Por lo tanto, las realizaciones no se limitan a ninguna combinación específica de circuitos de hardware y software. De hecho, los expertos en la materia deben entender que cualquier sistema informático adecuado puede implementar la funcionalidad descrita en este documento. El sistema informático puede incluir uno o una pluralidad de microprocesadores que trabajan para realizar las funciones deseadas. En una realización, las instrucciones ejecutadas por el microprocesador o microprocesadores son operables para hacer que los microprocesadores realicen las etapas descritas en este documento. Las instrucciones pueden almacenarse en cualquier medio legible por ordenador. En una realización, pueden almacenarse en una memoria de semiconductores no volátil externa al microprocesador o integrarse con el microprocesador. En otra realización, las instrucciones pueden almacenarse en un disco y leerse en una memoria semiconductor volátil antes de su ejecución por el microprocesador.

Aunque se han descrito ciertas realizaciones de la divulgación, estas realizaciones se han presentado a modo de ejemplo solo y no pretenden limitar el alcance de la divulgación. De hecho, los nuevos métodos, dispositivos y sistemas descritos en el presente documento pueden realizarse en varias otras formas. En consecuencia, el alcance de la presente divulgación se pretende que se defina solamente por referencia a las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un método implementado por ordenador para un nodo inductor (204) en un sistema informático distribuido para inducir un nodo inducido (206) en el sistema informático distribuido (601) para permitir que el inducido comunique mensajes a otros nodos en el sistema informático distribuido, comprendiendo el método implementado por ordenador:
 - generar una tarea de inducción que comprende la información necesaria para completar una inducción del nodo inducido en el sistema informático distribuido;
 - enviar (B21) la tarea de inducción al nodo inducido (206) a través de una red (1026);
 - recibir una solicitud de membresía (B25) del nodo inducido (206) a través de la red (1026), comprendiendo la solicitud de membresía información que identifica el nodo inducido e información suficiente para permitir la comunicación con el nodo inducido;
 - crear una membresía de arranque que define los roles del nodo inductor (204) y del nodo inducido (206), desplegar la membresía de arranque y enviar un mensaje de lista de membresía de arranque (B36) al nodo inducido;
 - crear (B34, B38) una máquina de estado determinista que hace referencia a la membresía de arranque creada; y recibir (B39) un acuse de recibo de que el nodo inducido ha creado una membresía de arranque correspondiente.
2. El método implementado por ordenador de la reivindicación 1, en el que la tarea de inducción comprende además al menos una tarea posterior a la inducción.
3. El método implementado por ordenador de las reivindicaciones 1 o 2, en el que la membresía de arranque comprende:
 - una identidad de membresía creada de manera determinista; y
 - roles del nodo inductor (204) y del nodo inducido (206).
4. El método implementado por ordenador de cualquier reivindicación anterior, que comprende además enviar al nodo inducido al menos un nodo y ubicaciones de las cuales el nodo inducido debe ser consciente.
5. El método implementado por ordenador de cualquier reivindicación anterior 1, en el que la generación se realiza con la tarea de inducción configurada para ser persistente al menos a través de reinicios del nodo inductor.
6. Un dispositivo informático (1000), que comprende:
 - una memoria (1004, 1006); y
 - un procesador (1002), estando configurado el procesador para ejecutar instrucciones almacenadas en la memoria para ejecutar el dispositivo informático como un nodo inductor (204) configurado para inducir un nodo inducido (206) en un sistema informático distribuido, estando configuradas las instrucciones almacenadas para hacer que el procesador:
 - genere una tarea de inducción que comprende la información necesaria para completar una inducción del nodo inducido en el sistema informático distribuido;
 - envíe (B21) la tarea de inducción al nodo inducido a través de una red (1026);
 - reciba (B26) una solicitud de membresía del nodo inducido a través de la red (1026), comprendiendo la solicitud de membresía información que identifica el nodo inducido e información suficiente para permitir la comunicación con el nodo inducido; cree una membresía de arranque que define los roles del nodo inductor (204) y del nodo inducido (206), despliegue la membresía de arranque y envíe (B36) un mensaje de lista de membresía de arranque al nodo inducido;
 - cree (B34, B38) una máquina de estado determinista que hace referencia a la membresía de arranque creada; y
 - reciba (B39) un acuse de recibo de que el nodo inducido (206) ha creado una membresía de arranque correspondiente.
7. Un medio de almacenamiento de datos tangible que almacena datos e instrucciones que configuran un dispositivo informático (1000) como un nodo inductor (204) configurado para inducir un nodo inducido (206) en un sistema informático distribuido (601), estando configurados los datos almacenados y las instrucciones para hacer que el dispositivo informático:
 - genere una tarea de inducción que comprende la información necesaria para completar una inducción del nodo inducido en el sistema informático distribuido;
 - envíe (B21) la tarea de inducción al nodo inducido a través de una red informática;
 - reciba (B26) una solicitud de membresía del nodo inducido a través de la red (1026), comprendiendo la solicitud de membresía información que identifica el nodo inducido e información suficiente para permitir la comunicación

- con el nodo inducido;
 cree una membresía de arranque que define los roles del nodo inductor (204) y del nodo inducido (206), despliegue la membresía de arranque y envíe (B36) un mensaje de lista de membresía de arranque al nodo inducido;
- 5 cree (B34, B38) una máquina de estado determinista que hace referencia a la membresía de arranque creada; y reciba (B39) un acuse de recibo de que el nodo inducido ha creado una membresía de arranque correspondiente.
8. Un método implementado por ordenador para que un nodo inducido (206) sea inducido en un sistema informático distribuido (601) por un nodo inductor (204), que comprende:
- 10 esperar la recepción de una tarea de inducción que comprende al menos un ticket de inducción;
 recibir la tarea de inducción y el ticket de inducción a través de una red informática y configurar una plataforma de aplicación utilizando información en el ticket de inducción;
- 15 enviar una solicitud de membresía de arranque al nodo inductor a través de la red informática, estando configurada la solicitud de membresía de arranque para informar al nodo inductor que el nodo inductor está iniciando la inducción; recibir, desde el nodo inductor, un mensaje de arranque listo para la membresía y
- 20 crear y desplegar una membresía de arranque; y
 crear (B34, B38) una máquina de estado determinista que hace referencia a la membresía de arranque y reconoce (B39) la pertenencia de arranque al nodo inductor.
9. El método implementado por ordenador de la reivindicación 8, en el que el envío se realiza con la solicitud de membresía de arranque que comprende una identidad de la tarea de inducción recibida e información que permite la comunicación con el nodo inducido.
- 25 10. El método implementado por ordenador de las reivindicaciones 8 o 9, que comprende además que la máquina de estado determinista reciba, desde el nodo inductor, una de:
- 30 a) una lista de ubicaciones y nodos que el nodo inducido debe tener en cuenta; y
 b) una propuesta que comprende una lista de ubicaciones y nodos que el nodo inducido debe tener en cuenta.
11. El método implementado por ordenador de cualquiera de las reivindicaciones 8 a 10, en el que la solicitud de membresía de arranque comprende al menos uno de una identidad de la tarea de inducción, nodo, ubicación, nombre de hospedador y puerto del nodo inducido.
- 35 12. El método implementado por ordenador de cualquiera de las reivindicaciones 8 a 11, en el que crear la membresía de arranque comprende crear la membresía de arranque con una identidad de membresía creada de manera determinista, y los roles de los nodos inductores e inducidos.
- 40 13. El método implementado por ordenador de cualquiera de las reivindicaciones 8 a 12, que comprende además realizar una tarea especificada en la tarea de inducción recibida.
14. Un dispositivo informático (1000), que comprende:
- 45 una memoria (1004, 1006); y
 un procesador (1002), estando el procesador configurado para ejecutar instrucciones almacenadas en la memoria para configurar el dispositivo informático como un nodo inducido (206) para ser inducido en un sistema informático distribuido (601) por un nodo inductor (204), estando configuradas las instrucciones almacenadas para hacer que el procesador:
- 50 espere la recepción de una tarea de inducción que comprende un ticket de inducción;
 reciba la tarea de inducción y el ticket de inducción a través de una red informática (1026) y configure una plataforma de aplicación utilizando información en el ticket de inducción;
- 55 envíe (B25) una solicitud de membresía de arranque al nodo inductor a través de la red informática, estando configurada la solicitud de membresía de arranque para informar al nodo inductor que el nodo inductor está iniciando la inducción;
 reciba, desde el nodo inductor, un mensaje listo para la membresía de arranque (B36) y
- 60 cree y despliegue una membresía de arranque; y
 cree una máquina de estado determinista que haga referencia a la membresía de arranque y confirme la membresía de arranque al nodo inductor.
15. Un medio de almacenamiento de datos tangible que almacena datos e instrucciones que configuran un dispositivo informático (1000) como un nodo inducido (206) para ser inducido en un sistema informático distribuido (601) por un nodo inductor (204), estando configuradas las instrucciones almacenadas para hacer que el dispositivo
- 65

informático (1000):

- 5 espere la recepción de una tarea de inducción que comprende un ticket de inducción;
reciba la tarea de inducción y el ticket de inducción a través de una red informática y configure una plataforma de aplicación utilizando información en el ticket de inducción;
- 10 envíe una solicitud de membresía de arranque al nodo inductor a través de la red informática, estando configurada la solicitud de membresía de arranque para informar al nodo inductor que el nodo inductor está iniciando la inducción;
reciba, desde el nodo inductor, un mensaje de arranque listo para la membresía y cree y despliegue una membresía de arranque; y cree una máquina de estado determinista que haga referencia a la membresía de arranque y confirme la membresía de arranque al nodo inductor.

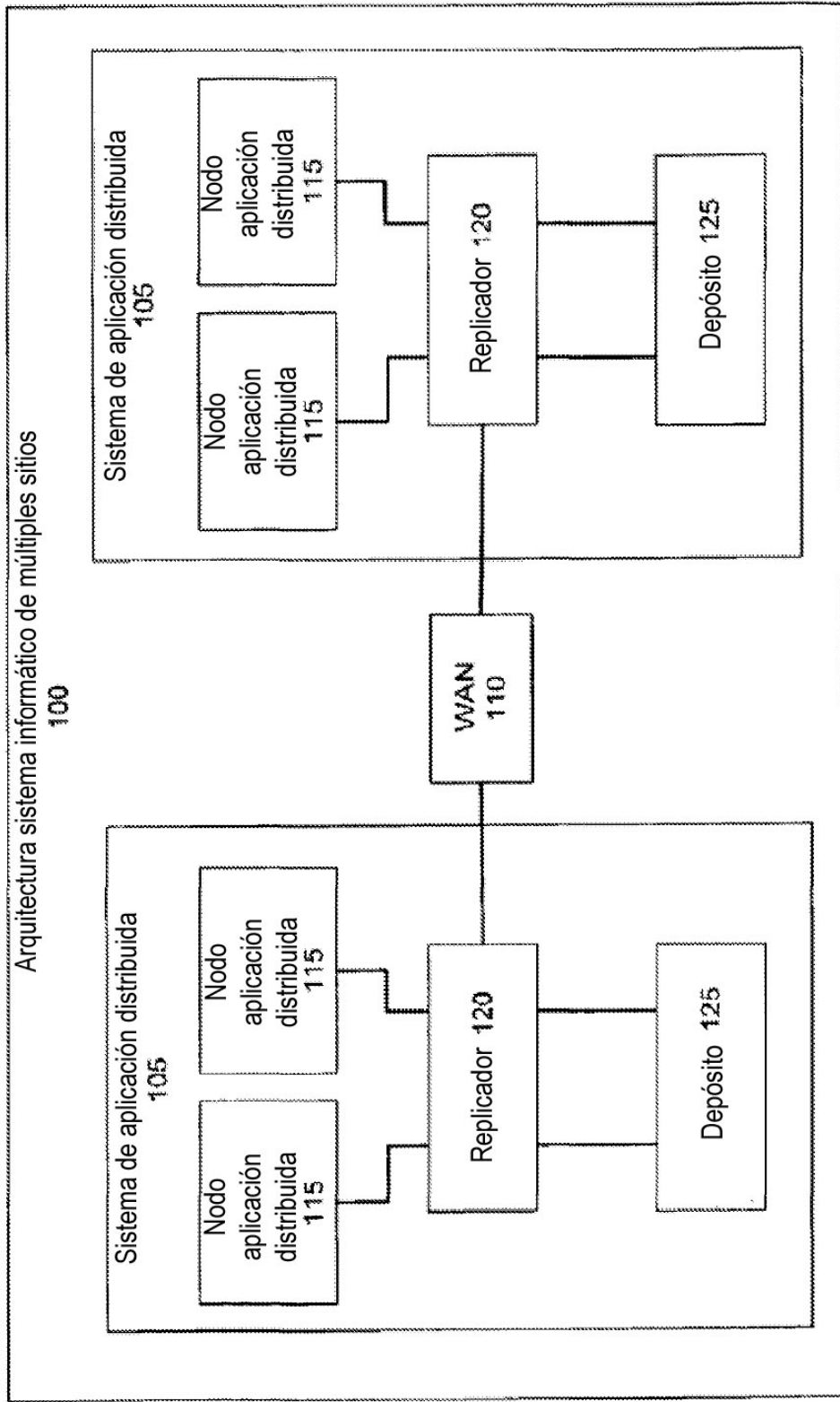


FIG. 1

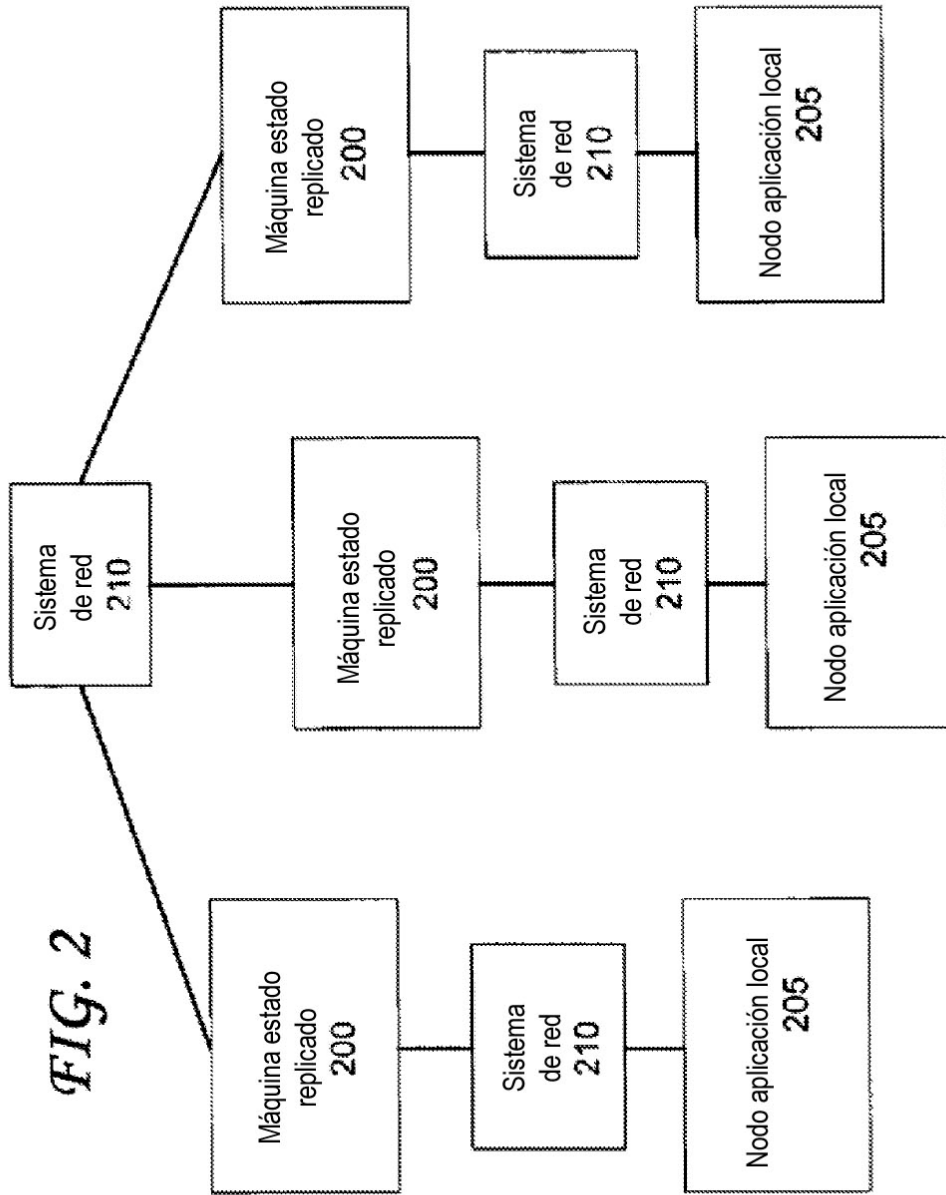


FIG. 3

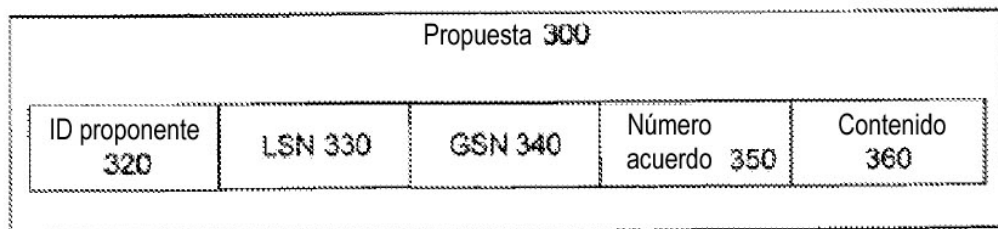
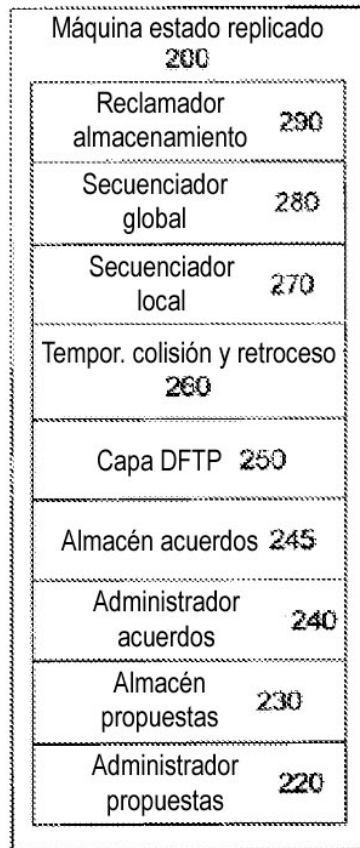


FIG. 4

Secuencia local 400				
ID proponente = 0x123	LSN	GSN	Número acuerdo	Contenido
ID proponente = 0x123	LSN	GSN	Número acuerdo	Contenido
ID proponente = 0x123	LSN	GSN	Número acuerdo	Contenido

FIG. 5

Secuencia global 500	
GSN #1	Gestión secuencia local 400
GSN #2	Gestión secuencia local 400
GSN #3	Gestión secuencia local 400
GSN #4	Gestión secuencia local 400

FIG. 6

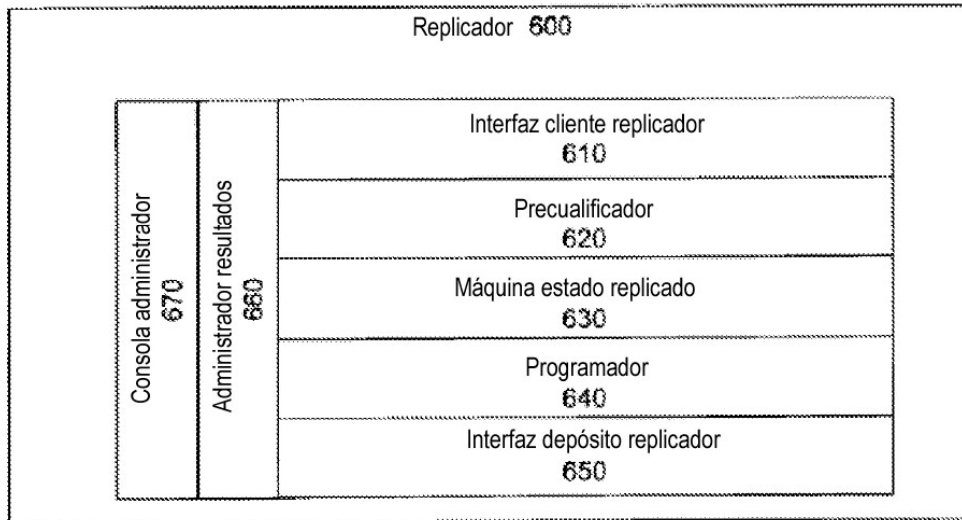


FIG. 7

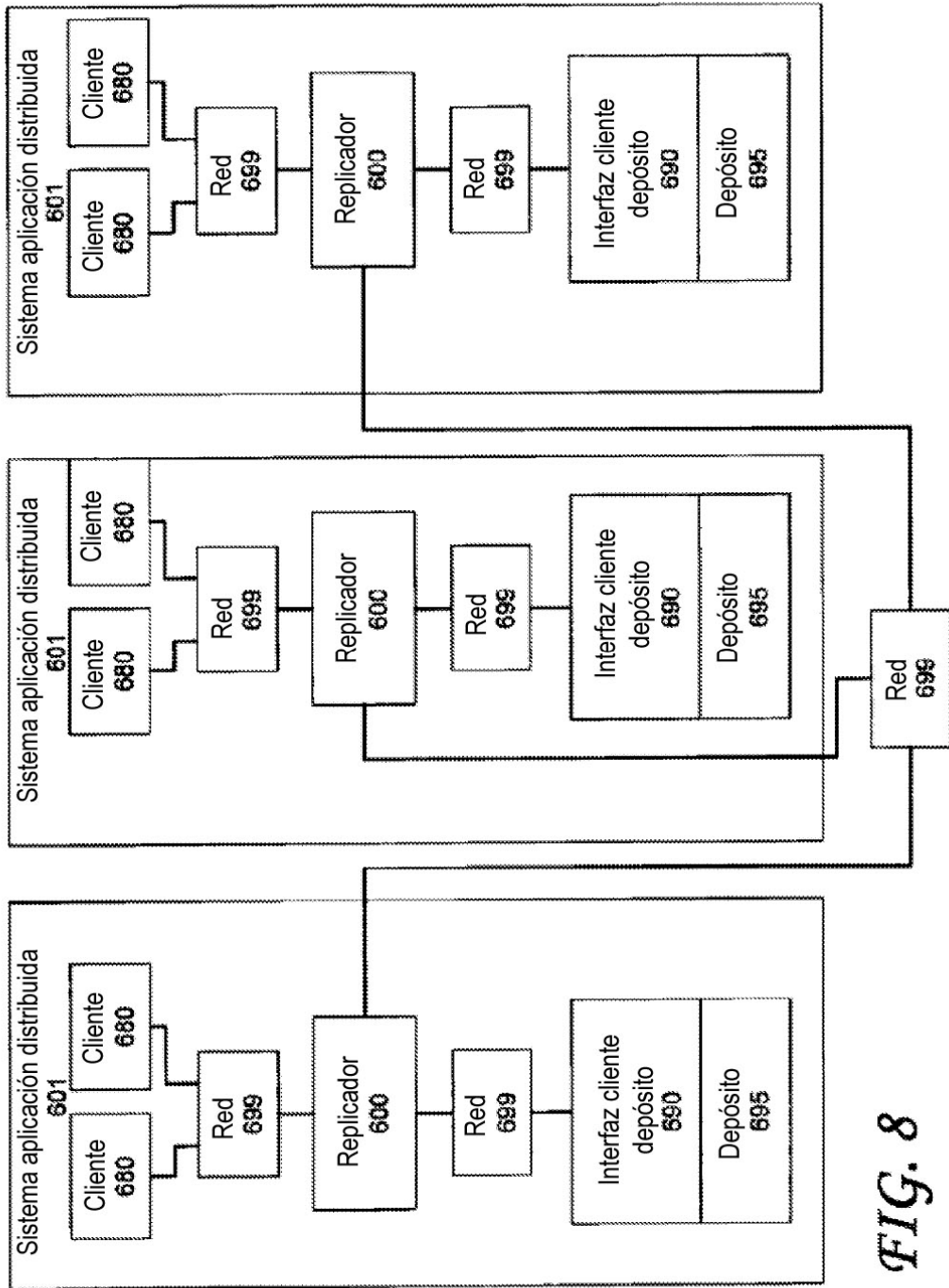


FIG. 8

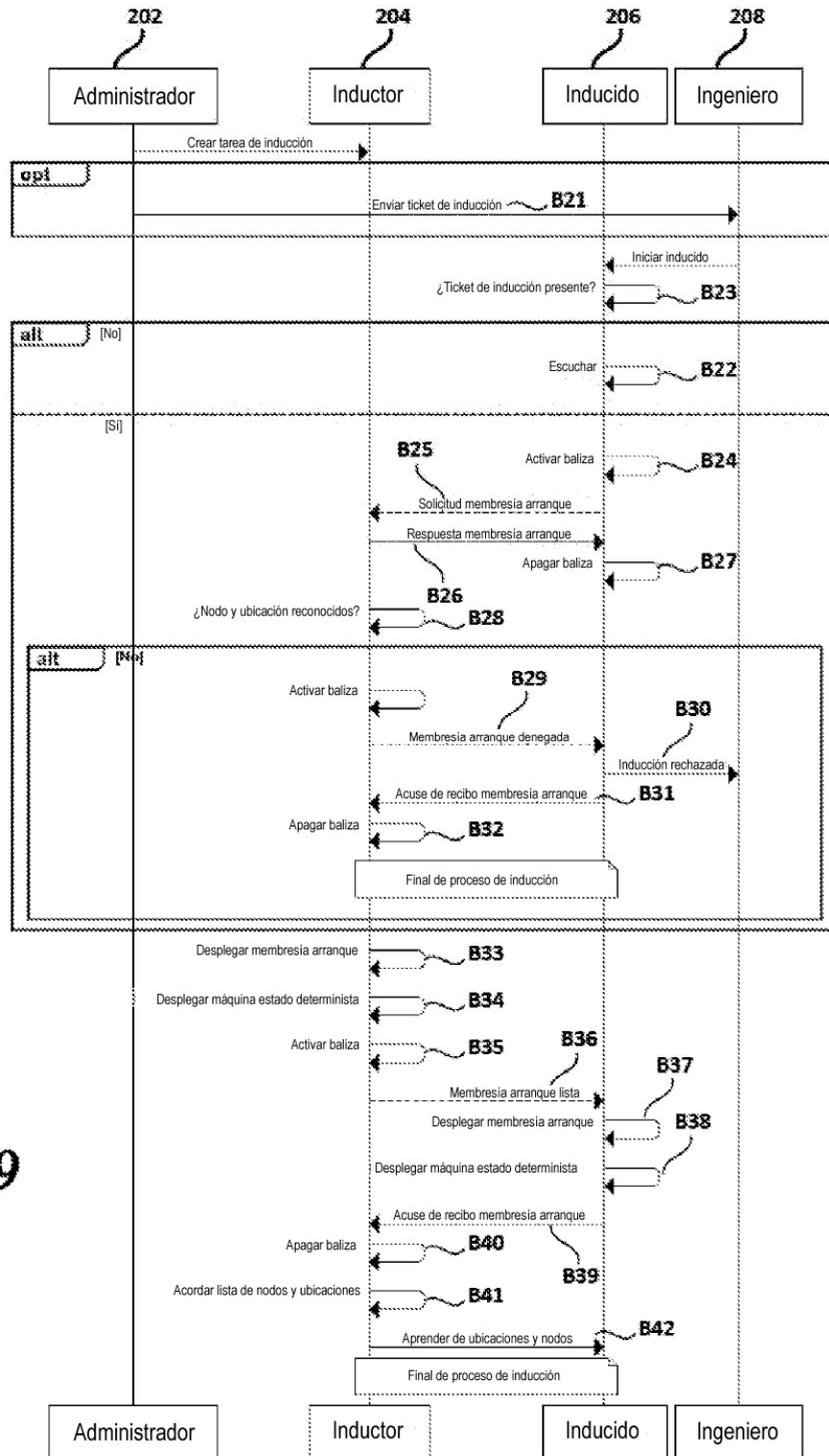


FIG. 9

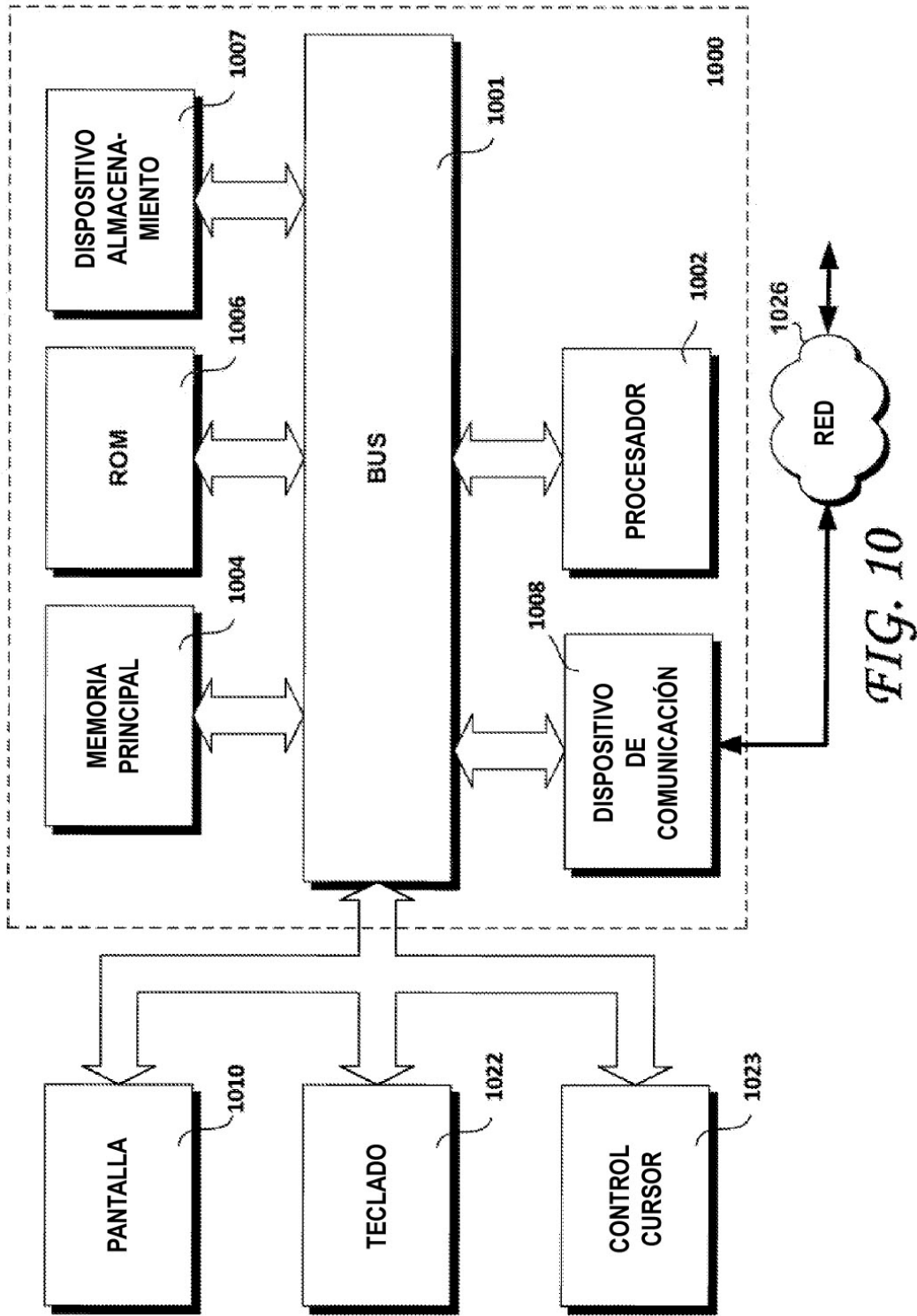


FIG. 10