(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 524 124**

(51) Int. CI.:

G06F 11/18 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **06.07.2012 E 12743101 (3)**

(97) Fecha y número de publicación de la concesión europea: **10.09.2014 EP 2689333**

(54) Título: **Método y sistema para almacenar y leer datos en o a partir de un almacenamiento de valor de clave**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
04.12.2014

(73) Titular/es:

NEC EUROPE LTD. (100.0%)
Kurfürsten-Anlage 36
69115 Heidelberg, DE

(72) Inventor/es:

DOBRE, DAN

(74) Agente/Representante:

ROEB DÍAZ-ÁLVAREZ, María

DESCRIPCIÓN

Método y sistema para almacenar y leer datos en o a partir de un almacenamiento de valor de clave

La presente invención se refiere a un método para almacenar datos en un almacenamiento de valor de clave que tiene una pluralidad de n servidores, en el que t < n servidores pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n.

La presente invención se refiere además a un método para leer datos almacenados en un almacenamiento de valor de clave que tiene una pluralidad de n servidores, en el que t < n servidores pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n.

La presente invención se refiere además a un sistema para almacenar datos en un almacenamiento de valor de clave que tiene una pluralidad de n servidores, en el que t < n servidores pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n y a un lector para leer datos almacenados en el almacenamiento de valor de clave, preferentemente para realizar un método de acuerdo con una de las reivindicaciones 1-13.

La presente invención se refiere además a un sistema para leer datos almacenados en un almacenamiento de valor de clave que tienen una pluralidad de n servidores en el que t < n servidores pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n y a un lector para leer datos almacenados en el almacenamiento de valor de clave, preferentemente para realizar un método de acuerdo con una de las reivindicaciones 1-13.

Los almacenamientos de valor de clave, también llamados almacenes de valor de clave (KVS) están obteniendo un interés creciente para varios sistemas distribuidos a gran escala que van desde bases de datos, motores de búsqueda, plataformas basadas en la nube, por ejemplo, marcos de programación en la nube como map-reduce, para aplicaciones colaborativas como las sociales redes. Por ejemplo, a menudo las bases de datos de almacenamiento convencionales implementan índices de búsqueda en la parte superior de un almacén de valor de clave distribuida para la escalabilidad y el rendimiento. Los almacenes de valor de clave no solo pueden servir como una capa de almacenamiento para las capas de nivel superior, sino que también pueden servir como aplicaciones directamente, tal como en el intercambio de archivos par a par. Si, por ejemplo, se bloquea uno o más de los servidores de almacenamiento debido a un error de software o de hardware o similar, el almacén de valor de clave compensará tal fallo. Los almacenes de valor de clave como Cassandra, Redis, HBase, Dynamo y Memcached toleran fallos o bloqueos de servidores de almacenamiento empleando la replicación.

Una característica deseable de los almacenes de valor de clave es garantizar la coherencia de los datos y la disponibilidad incluso en el caso de que se realice un acceso simultáneo de diferentes usuarios. El más alto grado de coherencia de los datos es la denominada coherencia atómica, lo que significa que si un cierto valor se almacena bajo una cierta clave o una operación de lectura devuelve el valor, entonces cada operación de lectura posterior devuelve el valor v correspondiente o un valor v' más nuevo pero nunca un valor que sea más antiguo que v. Los almacenes de valores de clave convencionales soportan la coherencia atómica a nivel de las claves individuales y el acceso transaccional a múltiples claves se realiza en capas de nivel superior. Para muchas aplicaciones, la coherencia atómica es obligatoria.

La disponibilidad es complementaria a la coherencia, es decir, si un cliente, que se supone que es un cliente correcto del almacén de valor de clave, recurre a una operación de lectura o escritura entonces la operación debería tener éxito a pesar de los servidores defectuosos e independientemente del comportamiento de otros clientes potencialmente defectuosos.

Para un número creciente de aplicaciones y sistemas, que dependen de un almacén de valor de clave como una capa de almacenamiento del núcleo, la robustez del almacén de valor de clave también es importante. Sin embargo, debido a la complejidad creciente de los sistemas distribuidos, que consisten de un gran número de nodos interconectados, la probabilidad de que falle algún nodo está aumentando. El riesgo de tales fallos accidentales de hardware y software, por ejemplo, los errores de software, también aumentan reduciendo significativamente la robustez del almacén de valor de clave. Un riesgo emergente adicional son las vulnerabilidades explotadas: Debido a la aparición de la computación en la nube, los datos se externalizan a las plataformas de servicio de la nube que son directamente accesibles desde internet. Una de las consecuencias son las vulnerabilidades explotables de las plataformas de servicio de la nube, por ejemplo, debido a los errores de software que resultan en unas potenciales pérdidas de datos y en la corrupción de los datos.

Para superar estos problemas, la literatura no de patente de James Hendricks, Gregory R. Ganger, Michael K. Reiter: "Low-overhead byzantine fault-tolerantstorage", SOSP 2007: 73-86 describe las operaciones de lectura necesarias para devolver solo en ejecuciones libres de contención para tolerar lectores bizantinos, lo que significa que pueden fallar de forma arbitraria.

En la literatura no de patente de Barbara Liskov, Rodrigo Rodrigues XP10927339: Tolerating Byzantine Faulty Clients In a Quorum System. Se usan las firmas digitales ICDCS 2006 para tolerar lectores bizantinos.

En la literatura no de patente de Amitanand S. Aiyer, Lorenzo Alvisi, Rida A Bazzi: “Bounded Wait-Free Implementation of Optimally Resilient Byzantine Storage Without (Unproven) Cryptographlc Assumptions", DISC 2007: 7-19, XP19100743 los canales de comunicación que se describen, entregan finalmente cada mensaje a todos los servidores, lo que en los sistemas asíncronos es difícil de implementar. Además, la latencia de lectura exhibida por el método descrito en la misma es lineal en el número de servidores.

La literatura no de patente de Dan Dobre et al: “Efficient Robust Storage Using Secret Tokens” 3 de noviembre de 2009 (03-11-2009), Estabilización, Seguridad y Seguridad de los Sistemas distribuidos, Springer Berlín. Heidelberg, Berlín, Heldelberg. Página(s) 269-283, XP019133325. ISN: 978-3-642-05117-3 muestra un almacenamiento robusto garantizando el progreso en todas las condiciones y nunca devuelve un valor antiguo ni un valor forjado usando testigos secretos.

Una de las desventajas es que los métodos convencionales mencionados anteriormente para los datos no autenticados no son prácticos debido a la alta latencia, la falta de escalabilidad y la falta de progreso en la contención. Además la mayoría de los métodos convencionales que no usan firmas están optimizados para un solo escritor.

Por lo tanto, es un objetivo de la presente invención proporcionar métodos y sistemas para almacenar y leer datos hacia o desde un almacenamiento de valor de clave que tolera fallos Bizantinos, en particular, mediante los servidores y los lectores.

Es un objetivo adicional que la presente invención proporcione métodos y sistemas para almacenar y leer datos hacia o desde un almacenamiento de valor de clave que sean una firma libre y proporcionen coherencia atómica.

Es incluso un objetivo adicional de la presente invención proporcionar métodos y sistemas para almacenar y leer datos hacia o desde un almacenamiento de valor de clave que sean escalables.

Es incluso un objetivo adicional de la presente invención proporcionar métodos y sistemas para almacenar y leer datos hacia o desde un almacenamiento de valor de clave que reduzcan la latencia para los datos de escritura y lectura.

Es un objetivo adicional de la presente invención proporcionar métodos y sistemas para almacenar y leer datos hacia o desde un almacenamiento de valor de clave que permita un mayor número de servidores pueden fallar de forma arbitraria.

Es incluso un objetivo adicional de la presente invención proporcionar métodos y sistemas para almacenar y leer datos hacia o desde un almacenamiento de valor de clave que sean más robustos que cualquier método o sistema de firma libre convencional.

Es incluso un objetivo adicional de la presente invención proporcionar métodos y sistemas para almacenar y leer datos hacia o desde un almacenamiento de valor de clave que permitan una fácil implementación con costes bajos y con una mayor flexibilidad con respecto a los datos que deben leerse o almacenarse.

De acuerdo con la invención, los objetivos mencionados anteriormente se consiguen mediante un método de la reivindicación 1.

De acuerdo con la reivindicación 1 el método para almacenar datos en un almacenamiento de valor de clave que tiene una pluralidad de n servidores, en el que $t < n$ servidores pueden fallar de forma arbitraria y en el que se cumple $3t+1 = n$.

De acuerdo con la reivindicación 1 el método se caracteriza por las etapas de

a) Generar una información de compromiso para una información secreta,
b) Difundir un primer mensaje, que incluye los datos que deben almacenarse, una clave correspondiente a los datos y la información de compromiso generada para los n servidores,
c) Almacenar la información incluida en el primer mensaje en al menos un número de servidores,
d) Proporcionar una primera información de confirmación de almacenamiento por al menos n-t servidores,
e) Difundir un segundo mensaje que incluye una clave correspondiente y la información secreta para los n servidores,
f) Almacenar la información incluida en el segundo mensaje, y
g) Proporcionar una segunda información de confirmación de almacenamiento por al menos n-t servidores.

De acuerdo con la invención, los objetivos mencionados anteriormente se consiguen además mediante un método de la reivindicación 2.

De acuerdo con la reivindicación 2 el método para leer datos almacenados en un almacenamiento de valor de clave que tiene una pluralidad de n servidores, en el que t < n servidores pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n.

De acuerdo con la reivindicación 2 el método se caracteriza por las etapas de

A) Difundir un primer mensaje que incluye una clave correspondiente a los datos que deben leerse
B) Recoger candidatos de los datos que deben leerse de al menos 2t+1 servidores,
C) Escribir de nuevo la información secreta correspondiente a la información de compromiso y a la información correspondiente a los datos que deben leerse,
D) Validar los candidatos recogidos en base a un emparejamiento de la información de compromiso y la información secreta,
E) Determinar los candidatos para los datos que deben leerse de acuerdo con los candidatos validados,
F) Seleccionar los datos que deben leerse en base a los t+1 mensajes de respuesta, que incluyen el mismo candidato de los datos que deben leerse y la información secreta correspondiente.

Los objetivos antes mencionados se consiguen además mediante un sistema de la reivindicación 14.

De acuerdo con la reivindicación 14, el sistema para almacenar datos en un almacenamiento de valor de clave que tiene una pluralidad de n servidores, en el que t < n servidores pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n, y un escritor para escribir los datos en el almacenamiento de valor de clave, preferentemente para realizarse con un método de acuerdo con una de las reivindicaciones 1-13.

De acuerdo con la reivindicación 14 el sistema se caracteriza por que el escritor está configurado para que pueda funcionar para difundir un primer mensaje que incluye los datos que deben almacenarse, una clave correspondiente para los datos y una información de compromiso, generados a partir de una información secreta, para los n servidores, por que

al menos un número de servidores está configurado para que pueda funcionar para almacenar la información incluida en el primer mensaje, por que
el escritor está configurado para que pueda funcionar para difundir un segundo mensaje que incluye una clave correspondiente y la información secreta para los n servidores después de recibir la primera información de confirmación de almacenamiento mediante al menos los n-t servidores, y por que
al menos los n-t servidores están configurados para que puedan funcionar para proporcionar una segunda información de confirmación de almacenamiento después de almacenar la información del segundo mensaje incluida en el segundo mensaje.

Los objetivos mencionados anteriormente se consiguen además mediante un sistema de acuerdo con la reivindicación 15.

De acuerdo con la reivindicación 15 el sistema para leer datos almacenados en un almacenamiento de valor de clave que tiene una pluralidad de n servidores, en el que t < n servidores pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n, y un lector para leer los datos almacenados en el almacenamiento de valor de clave, para realizarse con un método de acuerdo con una de las reivindicaciones 1-13.

De acuerdo con la reivindicación 15 el sistema se caracteriza por que el lector está configurado para que pueda funcionar para difundir un primer mensaje que incluye una clave correspondiente a los datos que deben leerse, para recoger los candidatos correspondientes a los datos que deben leerse a partir de al menos 2t+1 servidores, para escribir de nuevo la información secreta correspondiente a la información de compromiso, generada a partir de la información secreta y la información correspondiente a los datos que deben leerse, y para seleccionar los datos que deben leerse en base a los t+1 mensajes de respuesta, que incluyen el mismo candidato que debe leerse y la información secreta correspondiente, en el que los candidatos para los datos que deben leerse se han determinado de acuerdo con los candidatos validados en los que los candidatos recogidos se han validado en base a un emparejamiento de la información de compromiso y la información secreta.

De acuerdo con la invención, se ha reconocido en primer lugar que el método y los sistemas de acuerdo con las reivindicaciones 1, 2, 14 y 15 proporcionan un almacenamiento de valor de clave de firma libre robusto con una coherencia atómica cumplida y con tolerancia a fallos maliciosos de los servidores maliciosos, así como de los clientes maliciosos, es decir, se incluyen los lectores.

De acuerdo con la invención, se ha reconocido además en primer lugar que los métodos y los sistemas de acuerdo con las reivindicaciones 1, 2, 14 y 15 son escalables, es decir, todas las métricas relevantes, tales como la latencia, el número de mensajes, el tamaño del mensaje y los requisitos de almacenamiento de mensajes no dependen del tamaño de la población del cliente. No existe un límite superior en el número de clientes soportados de acuerdo con la presente invención. Además, los lectores pueden ser totalmente desconocidos.

De acuerdo con la invención, se ha reconocido además en primer lugar que la latencia de lectura se reduce significativamente y se degrada con gracia, es decir, se proporciona un máximo de dos rondas.

De acuerdo con la invención, se ha reconocido además en primer lugar que los métodos y los sistemas de acuerdo con las reivindicaciones 1, 2, 14 y 15 son ligeros de peso: Un cálculo de la información secreta y la información de compromiso es barato y no depende del valor que debe escribirse. Además la información secreta y la información de compromiso correspondiente pueden generarse por adelantado y/o fuera del sistema.

De acuerdo con la invención, se ha reconocido además en primer lugar que los sistemas y los métodos de acuerdo con las reivindicaciones 1, 2, 14 y 15 son robustos: Un lector nunca devuelve un valor olvidado o uno obsoleto. La disponibilidad está garantizada siempre y cuando se mantenga el secreto de los metadatos, es decir, se proporcione la información de compromiso.

De acuerdo con la invención, se ha reconocido además en primer lugar que los métodos y los sistemas de acuerdo con las reivindicaciones 1, 2, 14 y 15 no se basan en firmas digitales, de este modo, se ahorran costes computacionales asociados con la firma y la verificación de los datos. Además no son necesarios una infraestructura pública y/o distribuidores de confianza junto con la gestión de las claves asociadas.

Otras características, ventajas y realizaciones preferidas se describen en las siguientes reivindicaciones dependientes.

De acuerdo con una realización preferida, se genera y se asigna una información de marca de tiempo a los datos que deben almacenarse. Asignando la información de la marca de tiempo generada a los datos, los datos que deben almacenarse a partir de los escritores concurrentes pueden almacenarse fácilmente de acuerdo con la información de la marca de tiempo garantizando una manera fácil de coherencia atómica: Se asigna a los datos almacenados una cierta marca de tiempo y cuando se realiza una operación de lectura, la operación de lectura devuelve los datos correspondientes a la clave o datos más nuevos.

De acuerdo con una realización preferida adicional, una información de marca de tiempo generada es globalmente consistente. Por ejemplo, cuando los servidores están configurados para asignar localmente una marca de tiempo coherente globalmente a los datos, no es necesaria una coordinación explícita de los escritores con respecto a la información de la marca de tiempo: Por ejemplo, no es necesario un intercambio de su información de marca de tiempo, es decir, una coordinación de sus marcas de tiempo locales para obtener una información de marca de tiempo coherente, de esta manera se ahorran costes y se reduce la latencia.

De acuerdo con una realización preferida adicional, se recoge la información de la marca de tiempo generada antes de asignar la información de la marca de tiempo a los datos que deben almacenarse. Esto mejora la flexibilidad, ya que por ejemplo cada servidor puede crear o generar la información de la marca de tiempo local. A continuación, las diferentes marcas de tiempo se recogen y se coordinan por un escritor. Esto permite que cada servidor use su propia marca de tiempo local, que puede optimizarse para ciertas necesidades locales de los diferentes servidores.

De acuerdo con una realización preferida adicional, se evalúa la información de la marca de tiempo antes de realizar la etapa c) y/o f). Esto permite proporcionar los datos más actuales que están almacenados correspondientes a los datos con la marca de tiempo más alta.

De acuerdo con una realización preferida adicional, se verifica una validez de la información de la marca de tiempo, preferentemente intercambiando al menos simétricamente una información de marca de tiempo autenticada. Por ejemplo, cuando varios escritores con relojes no sincronizados actualizan la misma clave, necesitan determinar la marca de tiempo más alta vista por cualquier servidor correcto. Con este fin un escritor selecciona la marca de tiempo más alta vista por un servidor entre los n-t servidores. Los servidores maliciosos pueden responder con una marca de tiempo significativamente mayor que la última marca de tiempo de cualquier servidor correcto perdiendo con ello las marcas de tiempo. Para validar las marcas de tiempo los escritores pueden compartir una clave simétrica solo conocida por los escritores y autenticar cada marca de tiempo asignada a un valor con esa clave que proporciona la información de la marca de tiempo verificada.

De acuerdo con una realización preferida adicional, se genera una información de compromiso mediante el cálculo de clave (hashing). Una de las ventajas es, que debe incluirse poca información adicional en los mensajes y que la generación de la información de compromiso es barata. Además, el valor oculto puede estar disponible públicamente, permitiendo un desplazamiento de la verificación de los servidores a los lectores para un mejor rendimiento y/o escalabilidad. Para generar la información de compromiso mediante el cálculo de clave de una función de un solo sentido se aplica a una cadena de bits aleatoria de longitud suficiente. A continuación, se usa la información de compromiso en la etapa b) y la información secreta correspondiente representada por la cadena de bits aleatoria se difunde a continuación de acuerdo con la etapa e). En los servidores correctos se desencadena a continuación la etapa D) mediante una operación de lectura que incluye una comprobación de si un escritor ha comprometido la información secreta recibida de un lector.

De acuerdo con una realización preferida adicional, se genera una información de compromiso usando un valor aleatorio y un valor de polinomio de un polinomio aleatorio de grado t aplicada al valor aleatorio. Una de las ventajas es que la información secreta no puede construirse de forma prematura a partir del conocimiento colectivo de las partes maliciosas de al menos t partes y se evita que el adversario construya la información secreta correcta parcialmente. Información secreta correcta parcialmente significa información secreta que está validada por un subconjunto estricto de los servidores correctos. Para construir la información de compromiso se construye un polinomio P aleatorio de grado t, que representa la información secreta. A continuación, la n información de compromiso se construye escogiendo valores x_i aleatorios, uno para cada uno de los n servidores, y calculando P (x_i). El compromiso para el servidor i-enésimo consiste entonces del par (x_i, P (x_i)). A continuación, la información de compromiso puede difundirse de acuerdo con la etapa b) a través de los canales punto a punto garantizados o autenticados, garantizando que cada información de compromiso enviada a un servidor correcto se conoce solo por los destinatarios. A continuación, un escritor envía el polinomio P a todos los servidores de acuerdo con la etapa e).

Si se realiza una operación de lectura, la validación de acuerdo con la etapa D) en un servidor correcto se desencadena mediante la operación de lectura que consiste en comprobar si el escritor se ha comprometido con un polinomio P', es decir, si la curva que describe el polinomio P' asociado con un candidato representado por la tupla (k, ts, P') con la clave k, la marca de tiempo ts, y el polinomio P' recibido desde el lector contiene el punto (xi, P (x_i)) tomado de la tupla (k, ts, v, (x_i, P (x_i))) correspondiente recibida desde el escritor. La corrección puede garantizarse mediante los canales privados y los servidores correctos nunca deben revelar sus partes.

De acuerdo con una realización preferida adicional, se genera una información de compromiso que depende de los servidores, preferentemente para cada servidor se genera una información de compromiso separada correspondiente. Esto permite generar una información de compromiso local para cada servidor y permite proporcionar información teórica o la seguridad incondicional.

De acuerdo con una realización preferida adicional, se usan canales seguros para el intercambio de un mensaje y/o de información, preferentemente canales punto a punto autenticados. Esto garantiza que si un receptor recibe un mensaje de un remitente, entonces el remitente ha enviado el mensaje. Si el receptor y el remitente son ambos correctos, entonces cada mensaje enviado por el remitente se recibe finalmente por el receptor.

De acuerdo con una realización preferida adicional, los candidatos incluyen una clave almacenada y la información secreta más actualmente almacenada y/o más actualmente recibida. Esto permite una validación simple y rápida de los diferentes candidatos con una cierta marca de tiempo o más joven que representa la coherencia atómica.

De acuerdo con una realización preferida adicional, se transmite un conjunto de unión de todos los candidatos en la etapa C). Esto permite transmitir todos los candidatos con un mínimo de información en un mensaje, evitando que los candidatos idénticos se transmitan más de una vez. Otra ventaja adicional es, que puede realizarse una validación o por los servidores o posteriormente a los servidores por los lectores ya que toda la información necesaria para la validación se transmite a los servidores y puede ponerse a disposición de los lectores mediante los servidores.

De acuerdo con una realización preferida adicional, se proporcionan un tercer mensaje que incluye los candidatos recogidos de acuerdo con la etapa B) y un cuarto mensaje que incluye los candidatos validados de acuerdo con la etapa D) tras la recepción del primer mensaje. En particular, el tercer mensaje y el cuarto mensaje pueden proporcionarse en un solo mensaje.

Por ejemplo, un servidor correcto en respuesta a un primer mensaje de acuerdo con la etapa A) recoge candidatos para los datos que deben leerse de acuerdo con la etapa B) y además envía los candidatos validados de los candidatos recogidos. La condición de espera, es decir, recibir los t+1 segundos mensajes que incluyen el mismo candidato de los datos que deben leerse y la información secreta correspondiente, se aplica a continuación a los candidatos recogidos validados de acuerdo con la etapa D). Con el fin de evitar el bloqueo un lector espera a que la condición de espera se cumpla solo hasta una condición pre-proporcionada, por ejemplo, que se cumpla una expiración de un temporizador local. Si la operación de lectura en una fase no tiene éxito, el lector puede realizar la segunda fase mientras que espera en un hilo separado para completar la primera fase. Una de las ventajas es que la operación de lectura no se ve afectada con respecto a la latencia. Con el fin de evitar un reenvío de los datos, los servidores también pueden mantener la pista de los datos, preferentemente las tuplas de datos ya enviadas a un lector durante la primera fase de lectura.

De acuerdo con una realización preferida adicional, los candidatos de los datos que deben leerse se filtran tras recibir el cuarto mensaje. Una de las ventajas es que entonces no es necesaria la realización de una segunda fase ya que los lectores filtran los candidatos al final de la primera fase lo que permite una operación de lectura u obtención que se completa en la primera fase.

Existen varias maneras de cómo diseñar y desarrollar adicionalmente la enseñanza de la presente invención de una manera ventajosa. Para este fin, por una parte hay que hacer referencia a las reivindicaciones de patente subordinadas a la reivindicación 1 de patente y por la otra a la siguiente explicación de las realizaciones preferidas

de la invención a modo de ejemplo, ilustradas por las figuras. En relación con la explicación de las realizaciones preferidas de la invención mediante la ayuda de las figuras, en general, se explicarán las realizaciones preferidas y los desarrollos adicionales de la enseñanza.

En los dibujos

La figura 1 muestra un método para almacenar datos de acuerdo con una realización de la presente invención; y

La figura 2 muestra un método para leer datos de acuerdo con una realización de la presente invención.

La figura 1 muestra un método para almacenar datos de acuerdo con una realización de la presente invención.

En la figura 1 un escritor w realiza una operación put o de escritura put (k, v) para almacenar un valor v en una clave k en un almacenamiento de valor de clave que comprende cuatro servidores S1-S4 (en total n = 4 servidores y el número de servidores maliciosos permitidos t = 1). Se supone que la información commit de compromiso ya está generada de acuerdo con una información secreta proporcionada.

En un primer paso el escritor w difunde un primer mensaje 1a a los servidores S1-S4. En el primer mensaje 1a el escritor w incluye la clave k, el valor v que debe almacenarse, una información ts de marca de tiempo y la información commit de compromiso. El escritor envía el mensaje a todos los servidores S1-S4 y espera la respuesta de al menos n-t = 4-1 = 3 servidores.

Cuando una servidor S_i correcto recibe el primer mensaje 1a, el servidor S_i correspondiente que no ha recibido un mensaje con la clave k, el valor v, la información commit de compromiso y una información ts' de marca de tiempo con ts' > ts entonces el servidor S_i almacena (signo de referencia saving1) la información incluida en el primer mensaje 1a recibido, es decir, la clave k, el valor v, la información ts de marca de tiempo y la información commit de compromiso. En cualquier caso el servidor S_i responde al escritor w con un mensaje de reconocimiento, es decir, un mensaje de ok.

Después de recibir 4-1 = 3 mensajes de ok o respuestas, el escritor w revela la información secret secreta enviando una nueva ronda de mensajes (segundos mensajes, indicados con el signo de referencia 2a) que ahora incluyen la clave k, la marca de tiempo ts y la información secret secreta correspondiente a la información commit de compromiso. La clave k y la marca de tiempo ts se refieren a la información secreta, a la información de compromiso y al valor v correspondiente.

Cuando un servidor S_i correcto, es decir, que no ha fallado o no es malicioso, recibe el segundo mensaje 2a que incluye la clave k, la marca de tiempo ts y la información secret secreta, el servidor S_i almacena (saving2) la tupla <k, ts, secret> a menos que el servidor S_i haya recibido un segundo mensaje 2a adicional que incluye la clave k y la información de la marca de tiempo ts' con ts' > ts. En cualquier caso, el servidor S_i, responde con un mensaje 2b de ok. La operación put o de escritura que comprende una primera fase con los primeros mensajes 1a, 1b y una segunda fase con los segundos mensajes 2a, 2b se completa cuando un escritor w ha recibido 3 mensajes 2b de ok.

Conforme a los siguientes supuestos la latencia de la operación correcta o de put puede reducirse adicionalmente: La operación de put o correcta por un escritor se habilita para completarse en dos rondas de comunicación con los servidores.

Conforme a los supuestos de que a) el acceso a las distintas claves es secuencial, b) la red subyacente entre escritores/lectores y servidores es síncrona y c) que no existen fallos, la siguiente operación de put o de escritura puede realizarse de acuerdo con la invención. Los períodos que están representados por la sincronía, la contención y libre de fallo a menudo se consideran como o se llaman "el caso común".

En el caso común todos los servidores S_i correctos, aplican la misma secuencia de actualizaciones para la misma clave k y por lo tanto, todos mantienen la misma alta marca de tiempo ts. Por lo tanto, los servidores S_i, pueden asignar localmente una marca de tiempo ts coherente globalmente a un valor v sin coordinación explícita por un escritor w. El escritor puede omitir la recogida de la marca de tiempo y la fase de asignación. A continuación, la operación de put o de lectura se modifica como sigue: En la fase de pre-escritura el escritor w envía el primer mensaje 1a sin marca de tiempo ts; el primer mensaje 1a incluye, por lo tanto, la clave k, el valor de v, y la información commit de compromiso. El servidor S_i correcto al recibir el primer mensaje 1a aumenta la más alta marca de tiempo ts local a un valor aún más alto ts' = ts + 1 y almacena (saving1) la tupla recibida que incluye la clave k, el valor v y la información commit de compromiso con la marca de tiempo ts' = ts + 1.

A continuación, el servidor S_i responde al escritor w con un primer mensaje 1b que incluye la información de la marca de tiempo ts'. A continuación, el escritor w espera hasta recibir n-t = 4-1 = 3 respuestas de los servidores S1-S4 con iguales marcas de tiempo ts'. La conclusión es que t+1 = 2 servidores correctos han asignado al valor v la misma marca de tiempo ts'. Con el fin de no bloquear adicionalmente las operaciones de put o de escritura, el escritor w espera solo hasta una expiración de un temporizador local. En cualquier caso, el escritor w espera a

recibir las respuestas de n-t = 4-1 = 3 servidores. Cuando todas las respuestas (primeros mensajes 1b de ok) llevan la misma marca de tiempo ts, el escritor w procede con la fase de escritura enviando un segundo mensaje 2a que incluye una clave k, la marca de tiempo ts' y la información secret secreta. Si todas las respuestas 1b incluyen diferentes marcas de tiempo, el escritor w repite la fase de pre-escritura enviando un primer mensaje 1a' adicional que incluye una clave k, la marca de tiempo ts', un valor v y la información commit de compromiso con ts' = ts + 1 y con una marca de tiempo ts que representa la más alta marca de tiempo recibida.

La figura 2 muestra un método para leer datos de acuerdo con una realización de la presente invención.

En una operación get o de lectura un lector rd envía un primer mensaje 1a que incluye la clave k y espera respuestas desde los n-t = 4-1 = 3 servidores. Cuando un servidor S_i correcto, recibe un primer mensaje 1a que incluye la clave k desde el lector rd entonces el servidor S_i construye un conjunto de candidatos C_i que comprende la tupla con la marca de tiempo más alta de la clave k, la marca de tiempo ts y la información secret secreta recibida desde una operación de escritura anterior y, además, un conjunto de tuplas que incluye la clave k, la marca de tiempo ts' y la información secret' secreta con ts' > ts recibida desde otros y potencialmente maliciosos lectores rd. A continuación, el servidor S_i, envía el conjunto C_i, en un primer mensaje 1b correspondiente al lector rd.

Cuando el lector rd recibe los primeros mensajes 1b de respuesta desde 3 servidores, entonces el lector rd envía un segundo mensaje 2a a todos los n servidores S1-S4 que incluyen el conjunto C de unión de todos los conjuntos C_i candidatos. Cuando un servidor S_i correcto recibe un segundo mensaje 2a correspondiente que incluye el conjunto C de unión entonces el servidor S_i comprueba, para cada tupla que incluye una clave k, la marca de tiempo ts y la información secret secreta, si la tupla correspondiente que incluye la clave k, la marca de tiempo ts, el valor v y la información commit de compromiso se ha almacenado de forma local en el servidor S_i. Si es así el servidor S_i usa la información commit de compromiso para validar la información secret secreta. A continuación, el servidor S_i construye un conjunto V_i de tuplas que incluye la clave k, la marca de tiempo ts y el valor v que han pasado la comprobación de validez anteriormente mencionada. El servidor S_i, envía el conjunto V_i al lector rd en un segundo mensaje 2b de respuesta. En una última etapa el lector rd espera para recibir los mensajes 2b de respuesta correspondientes desde al menos 3 servidores. A continuación, se selecciona el valor v de los candidatos V_i validados, con la más alta marca de tiempo ts y se devuelve como el valor v para la operación get o de lectura get (k). Un candidato V validado que incluye la clave k, la marca de tiempo ts y la información secret secreta es válido cuando t+1 servidores responden con los segundos mensajes 2b de respuesta que incluyen la clave k, la marca de tiempo ts y el valor v. Un candidato con la clave k', la marca de tiempo ts', la información secret' secreta no es válido cuando un n-t = 4-3 = 1 servidor responde con los segundos mensajes 2b de respuesta que no incluyen la clave k', la marca de tiempo ts', el valor de v'.

En el caso común tal como se ha definido anteriormente la operación get o de lectura también puede omitir la segunda fase con los mensajes 2a, 2b si el candidato devuelto representado por la tupla con la clave k', la marca de tiempo ts, la información secret secreta con k' = k, más el correspondiente valor incluido en la tupla validada que incluye la clave k', la marca de tiempo ts, el valor v con k' = k se recopilan en la primera fase de n-t = 4-1 = 3 servidores en la primera fase (representada por los mensajes 1a y 1b). Un servidor S_i correcto, en respuesta a un primer mensaje 1a que incluye solo la clave k envía o responde de nuevo al lector rd con un primer mensaje 1b de respuesta y un segundo mensaje 2b de respuesta. Ambos mensajes 1b, 2b de respuesta incluyen la misma información como se representa en los mensajes 1b y 2b de la figura. 2. Esta separación de los mensajes es lógica y los datos incluidos pueden enviarse en un único mensaje físico.

En el lector rd, la condición de espera de la segunda fase de lectura representada por los mensajes 2a, 2b en la figura 2) se aplica a los mensajes 1b, 2b recibidos en la primera fase. La condición de espera significa que el lector rd espera para recibir los mensajes 1b, 2b de respuesta desde al menos n-t = 4-1 = 3 servidores.

Para evitar un bloqueo, el lector rd espera a que se cumpla la condición de espera únicamente hasta la expiración de un temporizador local. Además, el lector rd puede moverse a la segunda fase esperando en un hilo separado si la lectura en una fase no tiene éxito; por lo tanto la latencia de la operación get o de lectura no se ve afectada. Para evitar un reenvío de datos, los servidores pueden realizar un seguimiento de las tuplas de datos ya enviadas al lector rd durante la primera fase.

En resumen, la presente invención proporciona la disponibilidad, la integridad y la coherencia atómica: La integridad se cumple de la manera siguiente: Si una operación get o de lectura de una clave k devuelve un valor v, entonces el valor no se fabrica por un servidor bizantino. En detalle, la integridad se cumple porque si una operación get o de lectura recibe un valor v conforme a una clave k, entonces t+1 servidores han confirmado que se ha escrito el valor v. De esta manera, un servidor correcto ha almacenado el valor v, y por lo tanto v no está falsificado.

Además, se cumple la disponibilidad: Una operación put o de escritura de una clave k con un valor v y una operación get o de lectura de una clave k para un valor v nunca se bloquean.

Incluso se cumple además la coherencia atómica: Si una operación get o de lectura de una clave k devuelve un valor v entonces el valor v se escribe mediante la última operación put o de escritura con la clave k y el valor v anterior a)

una operación get o de lectura con la clave k o b) una operación put o de escritura concurrente con una operación get o de lectura. Si una operación get o de lectura con una clave k devuelve un valor v y una operación get o de lectura más tardía con un valor k' devuelve un valor v', entonces la operación de escritura o la operación put con la clave k y el valor v' no es anterior a la operación put o de escritura con la clave k y el valor v.

En detalle, la coherencia atómica se cumple por las siguientes razones: Si una operación put o de escritura put (k, v) es anterior a una operación get o de lectura get (k), entonces el candidato correspondiente que incluye la clave k, la información ts de la marca de tiempo y la información secret secreta (k, ts, secret) se almacena en t+1 servidores correctos. Ya que la operación get o de lectura para la clave k espera la respuesta de los n-t servidores, uno de los t+1 servidores correctos está entre los n-t servidores (intersección de quórum) y responde con el candidato (k, ts', secret') con ts' ≥ ts. Ya que el servidor es un servidor correcto, el candidato nunca se invalida y, finalmente, se convierte en válido. De este modo se cumple la coherencia atómica devolviendo el valor v' asociado.

Si las operaciones get o de lectura get (k) y get (k)' son operaciones de lectura de los lectores no maliciosos y si get (k) es anterior a get (k)' con v, v' serán los valores correspondientes devueltos y además se supone por contradicción que la operación put o de escritura con la clave k y el valor v' es anterior a la operación put o de escritura correspondiente con la clave k y el valor v entonces, ya que la operación get (k) devuelve el valor v-t+1, los servidores han respondido para el candidato correspondiente que incluye la clave k, la marca de tiempo ts y la información secret secreta. De este modo, se completa la fase de pre-escritura representada por los primeros mensajes 1a, 1b. Eso significa que t+1 servidores correctos han almacenado la clave k, la información ts de la marca de tiempo, el valor v y la información commit de compromiso.

La operación de lectura get (k) ha escrito de nuevo el candidato que incluye <k, ts, secret > a t+1 servidores correctos. Ya que la otra operación get o de lectura get (k)' espera la respuesta de los n-t servidores, al menos uno de ellos informa de la clave k, la marca de tiempo ts y la información secreta (k, ts, secret) y debido a que t+1 servidores correctos han almacenado (k, ts, v, commit)), entonces se valida finalmente el candidato (k, ts, secret). Ya que un lector rd siempre devuelve el mayor candidato válido con marca de tiempo y el valor v' se asigna a una marca de tiempo ts menor que v, porque la operación put put (k, v') es anterior a put (k, v), la operación get get (k)' no devuelve el valor de v' en contradicción con la suposición. Por lo tanto, se cumple además la coherencia atómica.

En resumen, la presente invención facilita que si un lector selecciona un valor como candidato devuelto, la operación de escritura debe haber completado la fase de pre-escritura, es decir, el valor se almacena en un conjunto de t+1 servidores correctos. Una operación de lectura puede omitir la contestación asociada a la fase de pre-escritura. La presente invención proporciona además la integridad de la fase de pre-escritura: El valor se escribe junto con la información de compromiso para una información secreta conocida únicamente por un escritor. La información secreta se revela por el escritor solo después de la finalización de la fase de pre-escritura. Durante una operación de lectura, la información de compromiso se usa para verificar la validez de la información secreta. Si se confirman tanto la validez de la información secreta como el valor por suficientes servidores, es decir, por al menos t+1, entonces se garantiza que el valor no se olvida y se completa la fase de pre-escritura.

De acuerdo con la presente invención, es suficiente una contestación que realiza una sola ronda. Además, la presente invención facilita que los lectores escriban de nuevo antes de determinar el candidato de vuelta actual. Por lo tanto, es suficiente la contestación de un conjunto que incluye el candidato de vuelta, dejando la tarea del filtrado para las lecturas subsiguientes. La contestación puede realizarse inmediatamente después de recoger todos los candidatos relevantes, eliminando la necesidad de fases de contestación separadas.

Además, la presente invención en el caso de la operación de lectura, escribe de nuevo solo los metadatos, es decir, los datos acerca del valor, pero no el valor en sí mismo, evitando lectores maliciosos que corrompan la información almacenada en los servidores y permitiendo los lectores bizantinos tolerados.

La presente invención facilita un secreto que se ha comprometido en una primera fase y un secreto que se revela en una segunda fase. Durante una operación de lectura, la información de compromiso se usa para comprobar la validez de la información secreta. Si una información secreta se valida por t+1 servidores entonces la operación put o de escritura ha realizado suficiente progreso y puede devolverse el candidato potencial. De lo contrario n-t servidores invalidan la información secreta y el candidato se descarta.

En una realización, la presente invención proporciona valores de pre-escritura sin una marca de tiempo y que tienen servidores de marcas de tiempo asignadas localmente sin una coordinación explícita. Esto permite una operación de lectura en una sola ronda. Los servidores responden a un primer mensaje con la clave k con ambos mensajes 1b y 2b como si ellos ya hubieran recibido un mensaje 2a desde el lector. A continuación, los lectores tienen que filtrar los candidatos al final de la primera ronda.

La presente invención proporciona

1) Un almacén de valor clave atómica libre de firmas robusto que tolera fallos maliciosos. Se soportan tanto los servidores como los clientes (lectores) maliciosos.

2) Escalabilidad en el sentido de que todas las métricas relevantes, tales como la latencia, el número de mensajes, el tamaño del mensaje y el requisito de almacenamiento no dependen del tamaño de la población de clientes. No existe un límite superior en el número de clientes. Además, los lectores pueden ser totalmente desconocidos. Escalabilidad significa que el número de mensajes y los tamaños de los mensajes no dependen de la cantidad de clientes. Debido a la falta de comunicación entre los servidores, el número de mensajes es proporcional al número de servidores.

3) Degradación óptima y con gracia de la latencia de lectura: una ronda en el caso común (representada por la sincronía, la contención y libre de fallo en una sola clave) y dos rondas en el peor de los casos. Baja latencia: La presente invención proporciona la más baja latencia posible de lectura de las dos rondas.

4) Un grado óptimo de replicación 3t+1; la fracción de servidores maliciosos soportados no se puede mejorar. La ventaja resultante es evidente en un entorno de centro de datos, donde un gran número de nodos de datos, cada uno manteniendo una parte de la fecha, se replican para la tolerancia a fallos.

5) Calcular los secretos y el compromiso de una manera barata y simple que no depende del valor que se está escribiendo. Por lo tanto, los metadatos pueden generarse por adelantado, fuera de la ruta crítica.

6) Garantizar la disponibilidad y la coherencia a pesar de un umbral de servidores maliciosos, cualquier número de clientes bloqueados, la asincronía y la contención. Un lector nunca devuelve un valor olvidado u obsoleto y la disponibilidad está garantizada siempre y cuando se mantenga el secreto de los metadatos. De esta manera, la seguridad del método y del sistema está siempre garantizada (determinísticamente). La presente invención es más robusta que cualquier protocolo basado en una firma.

Los protocolos de almacenamiento basados en firmas garantizan la coherencia e integridad solo en determinadas suposiciones, por ejemplo, el secreto de las claves, la intratabilidad de resolver ciertos problemas matemáticos (la dureza de la factorización). Cuando se violan estos supuestos, también se violan la coherencia y la integridad. La coherencia y la integridad se proporcionan mediante la presente invención. Incluso con el peor caso de desviación del conjunto de supuestos, se conserva la seguridad.

Muchas modificaciones y otras realizaciones de la invención expuesta en el presente documento vendrán a la mente del experto en la materia a la que pertenece la invención que tiene el beneficio de las enseñanzas presentadas en la descripción anterior y en los dibujos asociados. Por lo tanto, debe entenderse que la invención no está limitada a las realizaciones específicas divulgadas y que otras modificaciones y realizaciones están destinadas a incluirse dentro del alcance de las reivindicaciones adjuntas. Aunque se emplean términos específicos en el presente documento, se usan en un sentido genérico y descriptivo y no con fines de limitación.

REIVINDICACIONES

1. Un método para almacenar (put) datos (v) en un almacenamiento de valor de clave que tiene una pluralidad de n servidores (S_1, S_2, S_3, S_4), en el que t < n servidores (S_1, S_2, S_3, S_4) pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n,
caracterizado por las etapas de

a) Generar una información de compromiso (commit) para una información secreta (secret),
b) Difundir un primer mensaje (1a), que incluye los datos (v) que deben almacenarse, una clave (k) correspondiente a los datos (v) y la información de compromiso generada (commit) para los n servidores,
c) Almacenar ($saving_1$) la información incluida en el primer mensaje (1a) en al menos un número de servidores (S_1, S_2, S_3),
d) Proporcionar una primera información (1b) de confirmación de almacenamiento por al menos n-t servidores (S_1, S_2, S_3),
e) Difundir un segundo mensaje (2a) que incluye una clave (k) correspondiente y la información secreta (secret) para los n servidores (S_1, S_2, S_3, S_4),
f) Almacenar ($saving_2$) la información incluida en el segundo mensaje (2a), y
g) Proporcionar una segunda información (2b) de confirmación de almacenamiento por al menos n-t servidores (S_1, S_2, S_3).

2. Un método para leer (get) datos (v) almacenados en un almacenamiento de valor de clave que tiene una pluralidad de n servidores (S_1, S_2, S_3, S_4), en el que t < n servidores (S_1, S_2, S_3, S_4) pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n,
caracterizado por las etapas de

A) Difundir un primer mensaje (1a) que incluye una clave (k) correspondiente a los datos (v) que deben leerse
B) Recoger (1b) candidatos (C_2, C_i, $C_{antiguo}$) de los datos (v) que deben leerse desde al menos 2t+1 servidores (S_2, S_3, S_4),
C) Escribir de nuevo (2a) la información secreta (secret) correspondiente a la información de compromiso (commit) y a la información correspondiente a los datos (v) que deben leerse,
D) Validar (verification) los candidatos (C_2, C_i, $C_{antiguo}$) recogidos en base a un emparejamiento de la información de compromiso (commit) y la información secreta (secret),
E) Determinar los candidatos para los datos (v) que deben leerse de acuerdo con los candidatos (V_i) validados
F) Seleccionar los datos que deben leerse en base a los t+1 mensajes (2b) de respuesta, que incluyen el mismo candidato de los datos (v) que deben leerse y la información secreta correspondiente (secret).

3. El método de acuerdo con la reivindicación 1, caracterizado por que la información (ts) de marca de tiempo se genera y se asigna a los datos (v) que deben almacenarse.

4. El método de acuerdo con una de las reivindicaciones 1-3, caracterizado por que la información (ts) de marca de tiempo generada es consistente globalmente.

5. El método de acuerdo con una de las reivindicaciones 1, 3 y 4, caracterizado por que antes de asignar la información (ts) de marca de tiempo a los datos (v) que deben almacenarse, se recoge la información (ts) de marca de tiempo generada.

6. El método de acuerdo con una de las reivindicaciones 1, 3-5, caracterizado por que antes de realizar la etapa c) y/o la etapa f) se evalúa la información (ts) de marca de tiempo.

7. El método de acuerdo con una de las reivindicaciones 1-6, caracterizado por que se verifica una validación de la información (ts) de marca de tiempo, preferentemente intercambiando al menos simétricamente la información de marca de tiempo autenticada.

8. El método de acuerdo con una de las reivindicaciones 1-7, caracterizado por que la información de compromiso (commit) se genera mediante un cálculo de clave (hashing) y/o
la información de compromiso (commit) se genera usando un valor (x_i) aleatorio y un valor de polinomio de un polinomio (P) aleatorio de grado t aplicado al valor (x_i) aleatorio, en el que preferentemente
la información de compromiso (commit) depende del servidor (S_1, S_2, S_3, S_4), preferentemente para cada servidor (S_1, S_2, S_3, S_4) se genera una información de compromiso separada correspondiente (commit).

9. El método de acuerdo con una de las reivindicaciones 1-8, caracterizado por que los canales seguros se usan para el mensaje y/o el intercambio de información, preferentemente los canales punto a punto autenticados.

10. El método de acuerdo con las reivindicaciones 2-3, caracterizado por que los candidatos (C_2, C_i, $C_{antiguo}$) incluyen una clave (k) almacenada y la información secreta (secret) más recientemente almacenada y/o más recientemente recibida.

11. El método de acuerdo con la reivindicación 2, caracterizado por que un conjunto (C) de unión de todos los candidatos (C_2, C_i, $C_{antiguo}$) se transmite en la etapa C).

12. El método de acuerdo con una de las reivindicaciones 2, 10-11, caracterizado por que se proporcionan un tercer mensaje que incluye los candidatos (C_2, C_i, $C_{antiguo}$) recogidos de acuerdo con la etapa B) y un cuarto mensaje que incluye los candidatos (V_1, V_2, $V_{antiguo}$) validados de acuerdo con la etapa D) al recibir el primer mensaje (1a).

13. El método de acuerdo con la reivindicación 12, caracterizado por que los candidatos para los datos (v) que deben leerse se filtran al recibir el cuarto mensaje.

14. Un sistema para almacenar datos (v) en un almacenamiento de valor de clave que tiene una pluralidad de n servidores (S_1, S_2, S_3, S_4) en el que t < n servidores pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n, y un escritor (w) para escribir los datos (v) en el almacenamiento de valor de clave, para realizarse con un método de acuerdo con una de las reivindicaciones 1-13,
caracterizado por que
el escritor (w) está configurado para que pueda funcionar para difundir un primer mensaje (1a) que incluye los datos (v) que deben almacenarse, una clave (k) correspondiente para los datos (v) y una información de compromiso (commit), generados a partir de una información secreta (secret), para los n servidores (S_1, S_2, S_3, S_4), por que
al menos un número de servidores (S_1, S_2, S_3, S_4) está configurado para que pueda funcionar para almacenar la información incluida en el primer mensaje (1a), por que
el escritor (w) está configurado para que pueda funcionar para difundir un segundo mensaje (2a) que incluye una clave (k) correspondiente y la información secreta (secret) para los n servidores (S_1, S_2, S_3, S_4) después de recibir la primera información (1b) de confirmación de almacenamiento mediante al menos los n-t servidores (S_1, S_2, S_3, S_4), y por que
al menos los n-t servidores (S_1, S_2, S_3, S_4) están configurados para que puedan funcionar para proporcionar una segunda información (2b) de confirmación de almacenamiento después de almacenar la información del segundo mensaje (2a) incluida en el segundo mensaje (2a).

15. Un sistema para leer los datos almacenados en un almacenamiento de valor de clave que tiene una pluralidad de n servidores, en el que t < n servidores (S_1, S_2, S_3, S_4) pueden fallar de forma arbitraria y en el que se cumple 3t+1 = n, y un lector (rd) para leer los datos (v) almacenados en el almacenamiento de valor de clave, para realizarse con un método de acuerdo con una de las reivindicaciones 1-13,
caracterizado por que
el lector (rd) está configurado para que pueda funcionar para difundir un primer mensaje (1a) que incluye una clave (k) correspondiente a los datos (v) que deben leerse, para recoger los candidatos (C_2, C_i, $C_{antiguo}$) correspondientes a los datos (v) que deben leerse a partir de al menos 2t+1 servidores (S_1, S_2, S_3, S_4), para escribir de nuevo la información secreta (secret) correspondiente a la información de compromiso (commit), generada a partir de la información secreta (secret) y la información correspondiente a los datos (v) que deben leerse, y para seleccionar los datos (v) que deben leerse en base a los t+1 mensajes (C_2, C_i, $C_{antiguo}$) de respuesta que incluyen el mismo candidato (C_2, C_i, $C_{antiguo}$) que debe leerse y la información secreta (secret) correspondiente, en el que los candidatos (C_2, C_i, $C_{antiguo}$) para los datos (v) que deben leerse se han determinado de acuerdo con los candidatos (V_i) validados en los que los candidatos (C_2, C_i, $C_{antiguo}$) recogidos se han validado en base a un emparejamiento de la información de compromiso (commit) y la información secreta (secret).

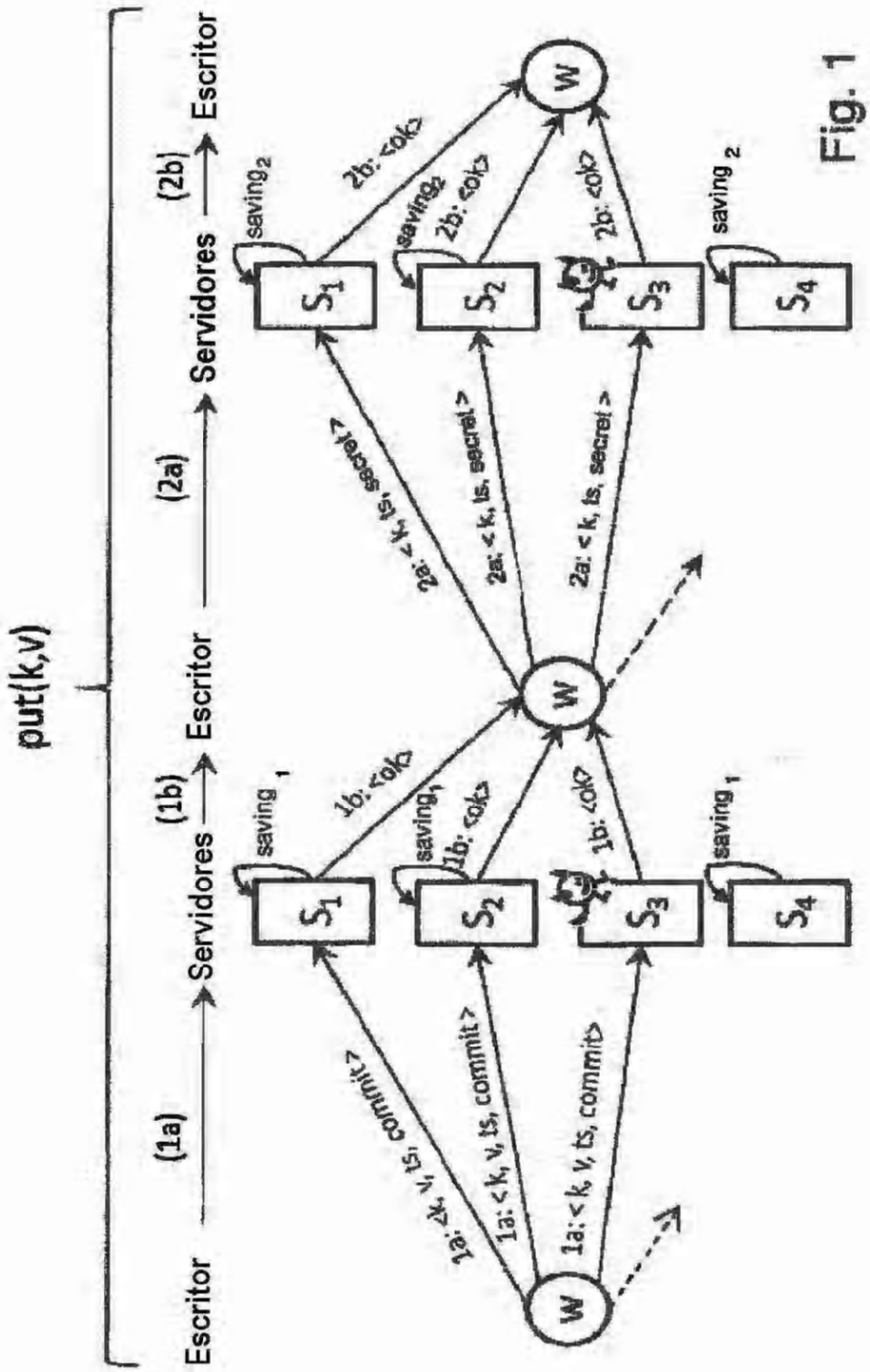
put(k,v)
Escritor
(1a)
Servidores
(1b)
Escritor
(2a)
Servidores
(2b)
Escritor
W
S1
S2
S3
S4
saving1
saving2
1a: <k, v, ts, commit>
1b: <ok>
2a: <k, ts, secret>
2b: <ok>

Fig. 1

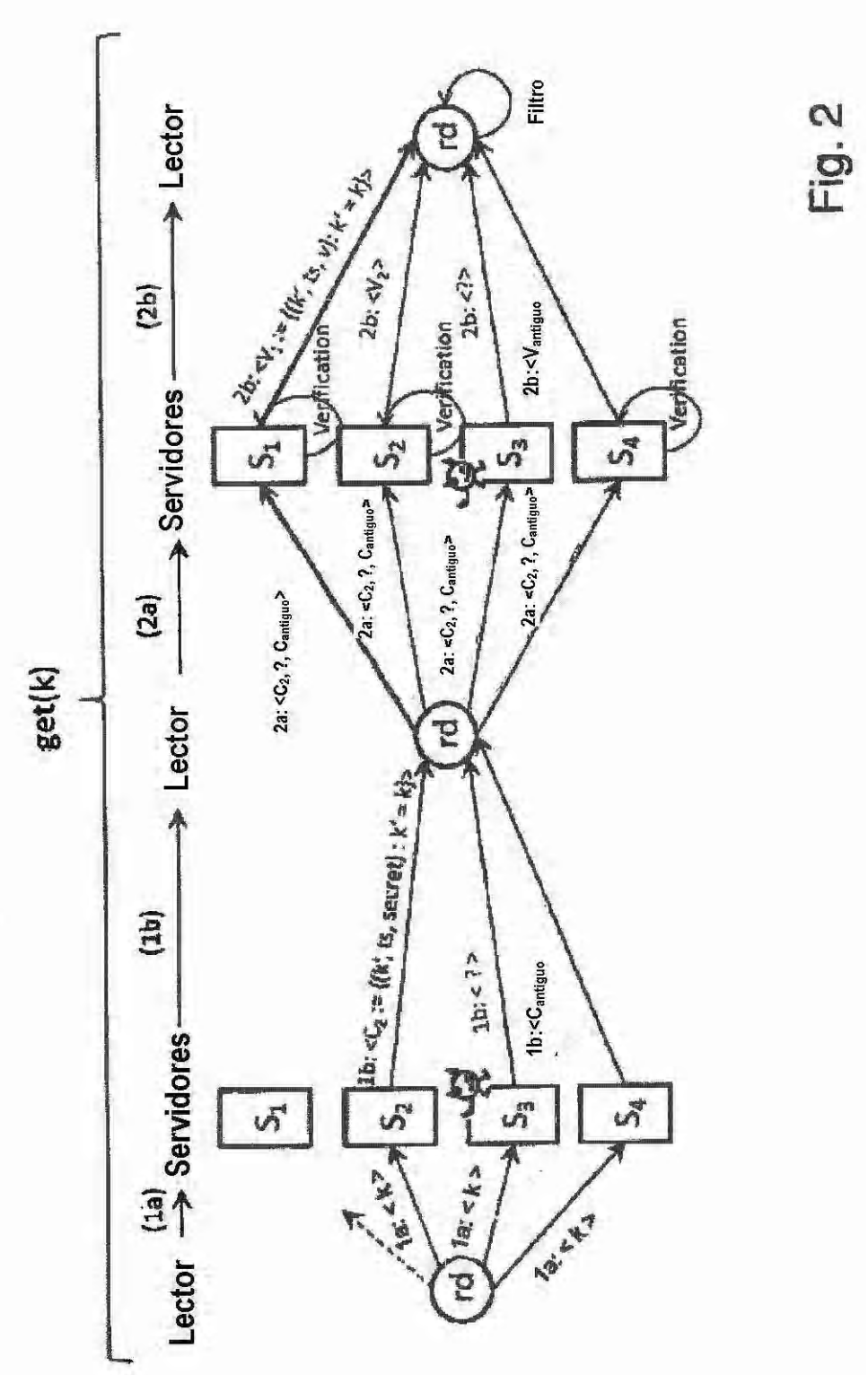
get(k)
Lector
(1a)
Servidores
(1b)
Lector
(2a)
Servidores
(2b)
Lector
S1
S2
S3
S4
rd
Filtro
Verification
1a: <k>
1b: <?>
1b:<Cantiguo
2a: <C2, ?, Cantiguo>
2b: <?>
2b:<Vantiguo

Fig. 2