

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 905 613**

51 Int. Cl.:

G06F 16/9032 (2009.01)

G06F 16/2455 (2009.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **31.07.2020** **E 20188834 (4)**

97 Fecha y número de publicación de la concesión europea: **01.12.2021** **EP 3771998**

54 Título: **Aprendizaje automático distribuido para la validez de datos en la memoria caché**

30 Prioridad:

31.07.2019 FR 1908757

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

11.04.2022

73 Titular/es:

AMADEUS S.A.S. (100.0%)
485 route du Pin Montard, Sophia Antipolis
06410 Biot, FR

72 Inventor/es:

GENTRIC, PHILIPPE FRANÇOIS GEORGES

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 905 613 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

Aprendizaje automático distribuido para la validez de datos en la memoria caché

La presente invención se refiere al procesamiento de consultas en un sistema de base de datos distribuido. Más específicamente, se refiere a mejorar la validez de los datos en la memoria caché mediante el aprendizaje automático.

5 El aumento de la validez de los datos en la memoria caché en entornos informáticos distribuidos ya ha dado lugar a una serie de enfoques y soluciones.

10 El documento US 2010/0106915 A1 describe el suministro de un sistema de notificación cliente/servidor basado en encuestas en una memoria caché distribuida para realizar un seguimiento de los cambios en los elementos de la memoria caché. Las memorias caché locales de los clientes utilizan el sistema de notificación para mantener los objetos locales sincronizados con un servicio de caché de fondo; y ajustar más dinámicamente el "alcance" de las notificaciones requeridas basándose en el número y la distribución de claves en la memoria caché local. El servidor mantiene los cambios y devuelve los cambios a los clientes que realizan el filtrado adecuado. Las notificaciones están asociadas con una sesión y/o una aplicación. Se emplea un sistema de inteligencia artificial para conocer nuevos eventos de notificación.

15 El documento US 2006/0271557 A1 describe el almacenamiento en la memoria caché de los resultados de las consultas de la base de datos, por lo que en un sistema de almacenamiento en la memoria caché se almacenan una o más consultas y los resultados se reciben de una base de datos en respuesta a dichas una o más consultas. Además, se monitoriza la información mantenida por dicha base de datos que indica las transacciones ejecutadas en dicha base de datos y se monitorizan los posibles cambios a dicha información mantenida por dicha base de datos. Se evalúa al menos una parte de dicha información almacenada por dicha base de datos para determinar si se han modificado partes de dicha base de datos. Se proporciona una notificación a dicho sistema de almacenamiento en la memoria caché si dicho acceso determina que se han modificado partes de dicha base de datos.

20 El documento US 2016/0125029 A1 describe los informes de Planificación de Recursos Empresariales (ERP) utilizando un servidor de memoria caché para almacenar en la memoria caché los resultados de consultas anteriores. La latencia de las consultas se reduce aún más al enrutar las consultas y las respuestas a las consultas a través del servidor de memoria caché en lugar de a través de la comunicación directa entre un dispositivo de consulta y un servidor que aloja la base de datos.

25 El documento US 8.117.153 B2 describe la gestión de una memoria caché de la base de datos distribuida. La memoria caché de la base de datos se distribuye sobre al menos dos sistemas de procesamiento de datos (subalmacén). Para las configuraciones que utilizan una base de datos de respaldo remota, se pueden configurar diferentes estrategias de sincronización entre un subalmacén y el almacén de respaldo remoto. El almacén de respaldo podría ser una base de datos de fondo o un subalmacén remoto (por ejemplo, en modo cliente/servidor, donde el cliente es una memoria caché para una base de datos en memoria). Estas estrategias se pueden aplicar en base a las solicitudes. La estrategia de sincronización se puede configurar para una transacción, una solicitud en particular o en una tabla o conjunto de tablas específico.

30 El documento WO 01/76192 A2 describe una arquitectura de red de borde distribuida en la que un centro de datos sirve como repositorio principal para el contenido cargado por los proveedores de contenido. Desde el centro de datos, el contenido se replica en todos o en un grupo seleccionado de sitios de punto de presencia (POP) "intermedios" geográficamente dispersos. Una pluralidad de sitios POP de borde se comunica con los sitios POP intermedios y sirven como memorias caché de red, almacenando contenido según lo soliciten los usuarios finales. En una realización, el administrador de distribución de contenido (CDM) implementa una política para administrar el espacio de la memoria caché en todos los servidores de archivos de borde utilizando datos de acceso a archivos almacenados en la base de datos central. Los archivos solicitados con relativa poca frecuencia y/o los archivos que no se han solicitado durante un período de tiempo relativamente largo en comparación con otros archivos pueden estar marcados como "para ser eliminados" desde el POP de borde.

35 El documento US 2016/0171008 A1 describe consultas de bases de datos de pre-computación y almacenamiento en la memoria caché y utiliza un modelo probabilístico para determinar las probabilidades de que los resultados de la consulta estén desactualizados.

40 Según un primer aspecto, se proporciona un método para procesar consultas en un sistema de base de datos distribuido. El sistema de base de datos distribuido comprende al menos una fuente de datos originales que almacena una cantidad de datos originales y una pluralidad de clientes que comprenden al menos un primer cliente y un segundo cliente. El primer cliente y el segundo cliente almacenan datos en la memoria caché que corresponden al menos a una parte de la cantidad de datos originales. El primer cliente y el segundo cliente alojan un modelo probabilístico que produce valores de validez asociados con los datos en la memoria caché que indican una probabilidad de que los datos en la memoria caché almacenados en el cliente correspondan a los datos originales correspondientes. El método comprende, en el primer cliente, seleccionar consultas aleatoriamente entre una pluralidad de consultas manejadas por el sistema de base de datos distribuido, en el momento en que se recibe la consulta respectiva de uno de la pluralidad de clientes. Para cada una de las consultas seleccionadas aleatoriamente, se recupera un primer fragmento

- de los datos en la memoria caché almacenados en el primer cliente y que corresponden a la consulta seleccionada aleatoriamente, y se recupera un primer fragmento de los datos originales que corresponden a la consulta seleccionada aleatoriamente de la al menos una fuente de datos originales. Para las consultas que no se seleccionan aleatoriamente, se recupera un segundo fragmento de los datos en la memoria caché almacenados en el primer cliente y que corresponden a la consulta y se evalúa el valor de validez del segundo fragmento de los datos en la memoria caché almacenados en el primer cliente. Si el valor de validez está por debajo de un umbral dado, se recupera un segundo fragmento de los datos originales que corresponden a la consulta de la al menos una fuente de datos originales, y se actualiza el segundo fragmento de los datos en la memoria caché almacenados en el primer cliente por el segundo fragmento de los datos originales. El modelo probabilístico del primer cliente se adapta en base al primer fragmento recuperado de los datos en la memoria caché y el primer fragmento recuperado de los datos originales utilizando un algoritmo de aprendizaje automático y se pone a disposición del segundo cliente.
- En algunas realizaciones, para consultas seleccionadas aleatoriamente, se recupera un primer fragmento adicional de datos en la memoria caché almacenados por al menos un cliente de la pluralidad de clientes y que corresponden a la consulta seleccionada aleatoriamente.
- En algunas realizaciones, el modelo probabilístico se adapta además en base al primer fragmento adicional de datos en la memoria caché.
- En algunas realizaciones, el método comprende además, en el segundo cliente, recibir al menos una parte del modelo probabilístico adaptado del primer cliente del primer cliente y adaptar el modelo probabilístico del segundo cliente en base al modelo probabilístico adaptado recuperado del primer cliente.
- En algunas realizaciones, poner a disposición del segundo cliente el modelo probabilístico adaptado comprende poner a disposición del segundo cliente un valor de validez actualizado para los primeros datos en la memoria caché asociados con un valor de ponderación, en donde el valor de ponderación indica un nivel de correlación del primer fragmento de los datos en la memoria caché almacenados en el primer cliente con otro fragmento de datos en la memoria caché almacenados en el segundo cliente, y en donde adaptar el modelo probabilístico del segundo cliente en base al modelo probabilístico adaptado recuperado del primer cliente comprende adaptar el modelo probabilístico del segundo cliente en base al valor de validez actualizado para los primeros datos en la memoria caché asociados y el valor de ponderación asociado.
- En algunas realizaciones, el método comprende además, en el primer cliente, recuperar al menos una parte del modelo probabilístico adaptado del segundo cliente del segundo cliente, en donde la adaptación del modelo probabilístico del primer cliente se basa además en el modelo probabilístico adaptado recuperado del segundo cliente.
- Según otro aspecto, se proporciona un ordenador, el ordenador que actúa como un primer cliente para manejar datos en un entorno informático distribuido que comprende una pluralidad de clientes que comprenden al menos un primer cliente y un segundo cliente y al menos una fuente de datos originales que almacena una cantidad de datos originales, estando dispuesto el ordenador para ejecutar el método de cualquiera de los aspectos antes mencionados.
- Según otro aspecto más, se proporciona un programa informático, comprendiendo el producto de programa informático instrucciones de código de programa almacenadas en un medio legible por ordenador para ejecutar las etapas del método según cualquiera de los aspectos antes mencionados, cuando dicho programa se ejecuta en un ordenador.
- Los presentes mecanismos se describirán con referencia a las figuras adjuntas. Los números de referencia similares generalmente indican elementos idénticos o funcionalmente similares.
- La Figura 1 es una representación esquemática de un entorno informático distribuido como se describe en la presente memoria.
- La Figura 2 es un diagrama de secuencia de mensajes para adaptar un modelo probabilístico para la validez de los datos en la memoria caché en un primer cliente.
- La Figura 3 es un diagrama de secuencia de mensajes según la Figura 2 y una recuperación adicional de datos en la memoria caché en un segundo cliente.
- La Figura 4 es un diagrama de secuencia de mensajes según la Figura 2 que incluye la adaptación de un modelo probabilístico de un segundo cliente en base a una actualización recibida del modelo probabilístico del primer cliente.
- La Figura 5 es un diagrama de secuencia de mensajes para adaptar el modelo probabilístico del primer cliente utilizando partes del modelo probabilístico adaptadas por el segundo cliente.
- La Figura 6 muestra funciones ejemplares de la probable precisión que disminuye con el tiempo según algunas realizaciones.
- La Figura 7 es una representación esquemática de los componentes internos de un ordenador según algunas realizaciones.

La descripción en cuestión generalmente se refiere al manejo de consultas en un sistema de base de datos. El término "consulta" incluye todos los tipos de solicitudes de base de datos, incluyendo, por ejemplo, leer solicitudes para recuperar datos y escribir solicitudes para insertar, cambiar o eliminar datos.

5 Una plataforma de búsqueda puede mantener resultados de búsqueda recopilados previamente, procesados previamente y/o calculados previamente en una base de datos, también denominados datos en la memoria caché en lo sucesivo. La plataforma de búsqueda recibe consultas tales como consultas de búsqueda, procesa cada una de las consultas de búsqueda recibidas de un cliente y realiza las búsquedas correspondientes dentro de los resultados de búsqueda recopilados previamente o calculados previamente. La plataforma de búsqueda puede enviar una solicitud de búsqueda para realizar una búsqueda para cada consulta de búsqueda a la base de datos, también denominada fuente de memoria caché en lo sucesivo. La solicitud de búsqueda incluye al menos parte de uno o más criterios de búsqueda o parámetros de la consulta. La fuente de la memoria caché ejecuta la búsqueda en base a la solicitud de búsqueda, es decir, busca datos entre los datos en la memoria caché que corresponden a los criterios de búsqueda o los parámetros de la solicitud de búsqueda y envía los datos en la memoria caché encontrados como resultados de búsqueda calculados previamente o recopilados previamente de vuelta a la Plataforma de búsqueda. Los resultados de búsqueda calculados previamente o recopilados previamente (es decir, datos en la memoria caché) tienen valores de parámetro que corresponden a los criterios de búsqueda o los parámetros de la solicitud de búsqueda. La plataforma de búsqueda puede devolver al cliente desde el que se recibió la consulta de búsqueda cualquier resultado de búsqueda correspondiente, calculado previamente o recopilado previamente, de la fuente de la memoria caché.

20 En caso de que no se encuentren resultados de búsqueda correspondientes calculados previamente o recopilados previamente en la fuente de la memoria caché, la plataforma de búsqueda puede hacer que los criterios de búsqueda de la consulta de búsqueda sean menos restrictivos y puede realizar una búsqueda revisada enviando solicitudes de búsqueda revisadas con los criterios de búsqueda menos restrictivos a la fuente de la memoria caché. La plataforma de búsqueda también puede realizar la búsqueda revisada no aplicando todos los criterios de búsqueda dados, sino solo un subconjunto de los criterios de búsqueda. Los detalles de cuáles y cómo los criterios de búsqueda de las consultas de búsqueda se hacen menos restrictivos, y/o qué criterios de búsqueda o parámetros son menos importantes en las consultas de búsqueda, pueden establecerse mediante reglas dadas o determinarse utilizando técnicas de aprendizaje automático, lo último dando como resultado, por ejemplo, modelos probabilísticos basados en criterios de búsqueda históricos y/o resultados de búsqueda calculados previamente o recopilados previamente.

30 El Aprendizaje Automático como una rama de la Inteligencia Artificial posee la capacidad de rastrear y evaluar de manera flexible los entornos cambiantes. Por lo tanto, los sistemas de Aprendizaje Automático se emplean, por ejemplo, para establecer y mantener un modelo probabilístico para aproximar la validez de los datos en la memoria caché, con el fin de mantener los datos en la memoria caché válidos mediante el empleo de recursos de cálculo de manera eficiente. Un ejemplo lo da la solicitud de Patente Europea número 19 166 538. Se aplican distintos algoritmos y técnicas para los sistemas de aprendizaje automático, tal como el aprendizaje supervisado, semisupervisado y no supervisado, el aprendizaje por refuerzo, el aprendizaje de características, el aprendizaje escaso de diccionario, la detección de anomalías, etc.

40 Los sistemas de aprendizaje automático a menudo necesitan procesar grandes cantidades de datos de entrenamiento hasta que su modelo subyacente converja y pueda utilizarse de manera efectiva en un sistema productivo. En algunos campos, tales como las telecomunicaciones, la seguridad pública, los seguros de salud y el comercio minorista, incluso se requerirá "Big Data" en el intervalo de Tera- y Zettabytes para extraer, por ejemplo, modelos y patrones de usuario significativos. Por lo tanto, los sistemas de aprendizaje automático de última generación requieren una alta potencia de cálculo, una mayor capacidad de almacenamiento y una mayor capacidad de memoria caché en el sistema informático que aloja el sistema de aprendizaje automático. Además, puede ocurrir una cantidad significativa de transmisiones de red cuando el sistema de aprendizaje automático está alojado por ejemplo, en una solo ordenador, tal como un servidor central que forma parte de un entorno informático distribuido.

Otro aspecto es que un sistema de aprendizaje automático único y operado de forma centralizada constituye un único punto de fallo. Cuando no hay disponible un sistema alternativo de aprendizaje automático, en caso de fallo, la funcionalidad del entorno informático distribuido puede verse gravemente afectada.

50 Con el fin de reducir la cantidad de datos a transmitir, así como la pérdida general de la funcionalidad de aprendizaje automático en caso de fallo, en la presente memoria se enseña la recopilación de datos descentralizada aplicada en un sistema de aprendizaje automático distribuido, haciendo uso de este modo de la capacidad de replicación de datos de un sistema entre iguales.

55 La presente descripción se refiere generalmente al almacenamiento y la puesta a disposición de los datos en la memoria caché de una manera distribuida, es decir, en una pluralidad de nodos de red de un sistema entre iguales. Generalmente, la pluralidad de nodos de red, también denominados clientes en lo sucesivo, almacenan copias, datos calculados previamente, procesados previamente o preparados de cualquier forma (es decir, los datos en la memoria caché como se introdujo anteriormente) correspondientes a los datos originales. Los datos originales se mantienen en una fuente 3 de datos originales (que podría ser un sistema distribuido por sí solo). Los datos almacenados en un sistema entre iguales se pueden distribuir entre varios clientes del sistema entre iguales de varias formas. En un caso, cada cliente tiene datos en la memoria caché que están separados con respecto a los datos en la memoria caché

mantenidos por los otros clientes de la red entre iguales. Un cliente de la red no comparte los mismos datos con otro cliente. Por ejemplo, varios registros de datos pueden identificarse y recuperarse mediante varios valores-clave asociados X_1, X_2, \dots, X_n puede ser almacenado por tres clientes, denominados primer cliente, segundo cliente y tercer cliente. El número de registros de datos forma al menos una parte de los datos en la memoria caché que corresponde a los datos originales de la fuente de datos originales. El primer cliente puede almacenar los valores-clave X_1, X_2, \dots, X_n , el segundo cliente almacena los valores-clave X_{n+1}, \dots, X_m y el tercer cliente almacena los valores-clave X_{m+1}, \dots, X_n . Por tanto, en tal modelo de almacenamiento, ningún registro de datos de los datos en la memoria caché se almacena en más de un cliente de caché.

En otro caso, todos los clientes de la red tienen los mismos datos, por ejemplo, todos los datos en la memoria caché correspondientes a los datos originales. Con referencia al ejemplo mencionado anteriormente, el primer, segundo y tercer cliente almacenan el mismo registro de datos identificado por los valores-clave X_1, X_2, \dots, X_n .

Cabe señalar que en el ejemplo mencionado anteriormente, no todos los clientes del primero, el segundo y el tercer cliente pueden tener una versión actualizada de los registros de datos en la memoria caché, ya que los datos originales subyacentes pueden cambiar de vez en cuando en la fuente 3 de datos originales y, por lo tanto, los datos en la memoria caché en poder de un cliente pueden estar desactualizados, mientras que el otro cliente puede tener una versión actualizada del registro de datos en la memoria caché.

En la práctica, se pueden emplear modelos de almacenamiento de datos mixtos donde los datos en la memoria caché distribuidos entre los distintos clientes se superponen parcialmente. En el ejemplo mencionado anteriormente de los tres clientes que forman parte de un entorno informático distribuido, el primer cliente puede, por ejemplo, almacenar los registros de datos en la memoria caché con los valores-clave X_1, X_2, \dots, X_n , el segundo cliente puede almacenar $X_{n-2}, X_{n-1}, X_n, X_{n+1}, \dots, X_m$, el tercer cliente puede almacenar los valores-clave $X_{m-2}, X_{m-1}, X_m, X_{m+1}, \dots, X_n$.

Además, los datos en la memoria caché almacenados en distintos clientes pueden tener un nivel de correlación entre sí. El nivel más alto de correlación es la identidad. Por ejemplo, el registro de datos en la memoria caché con el valor-clave X_2 almacenado en el primer cliente se correlaciona con el registro de datos en la memoria caché con el mismo valor-clave X_2 almacenados en el segundo cliente (independientemente de si ambas versiones del registro X_2 de datos en la memoria caché están actualizados, es decir, en línea con los datos originales subyacentes a X_2). Por lo tanto, si los datos originales correspondientes a los datos X_2 en la memoria caché se cambian, ambos registros X_2 de datos en la memoria caché tienen la misma probabilidad de quedar desactualizados por este cambio. A un nivel más bajo de correlación, los registros de datos en la memoria caché pueden correlacionarse debido al principio conocido de localidad, por ejemplo, ya que ellos (o los datos originales subyacentes) son "vecinos" en la base de datos de la fuente 3 de datos originales. Por ejemplo, dado que el registro de datos en la memoria caché con el valor-clave X_2 está estrechamente correlacionado con el registro X_3 de datos en la memoria caché vecinos, es probable que una invalidación de X_2 (debido a un cambio de los datos originales subyacentes) también representa X_3 inválido. Además, también puede haber registros de datos en la memoria caché que solo tengan una correlación muy baja o que no tengan correlación alguna. Para tales registros de datos en la memoria caché no correlacionados, una alta probabilidad de que un primer registro de datos en la memoria caché esté desactualizado no tiene ninguna implicación si un segundo registro de datos que no está correlacionado con el primer registro de datos en la memoria caché también está desactualizado.

La presente descripción se refiere a un sistema de aprendizaje automático distribuido aplicado en un sistema entre iguales según el cual una pluralidad de clientes no solo tiene una cantidad de datos en la memoria caché, sino que un cliente también tiene un modelo probabilístico respectivo que modela la probabilidad de validez de los datos en la memoria caché almacenados en el cliente. La entidad de aprendizaje automático en un cliente puede adaptarse al modelo de almacenamiento de los datos en la memoria caché entre los clientes, es decir, un modelo de almacenamiento disyuntivo, un modelo de "datos almacenados en múltiples clientes" o un modelo de almacenamiento particularmente superpuesto. Más específicamente, un cliente de la red entre iguales que tiene una fracción particular de los datos distribuidos puede tener información sobre cuáles de los otros clientes de la red tienen la misma o partes de la fracción particular de los datos y/o datos en la memoria caché que se correlacionan con los propios datos en la memoria caché. Las entidades de aprendizaje automático en los clientes intercambian información acerca de las adaptaciones del modelo probabilístico local, opcionalmente teniendo en cuenta la correlación de sus datos en la memoria caché.

La Figura 1 ilustra un sistema informático distribuido como se utiliza en la presente memoria de una manera esquemática que comprende al menos un primer cliente 1, un segundo cliente 2, una fuente 3 de datos originales y una serie de interfaces 4 de comunicación. La fuente 3 de datos originales puede estar compuesta por varias fuentes de datos originales individuales.

El primer cliente 1, el segundo cliente 2 y la fuente 3 de datos originales pueden estar constituidos por varios ordenadores con hardware dependiendo de los requisitos de rendimiento. Tanto el primer cliente 1, como el segundo cliente 2 como la fuente 3 de datos originales están incorporados, por ejemplo, como ordenadores con hardware estacionario o móvil que comprenden ordenadores 100 como se ilustra en la Figura 7, y/o como sistemas especializados tales como sistemas integrados dispuestos para un propósito técnico particular, y/o como componentes

de software que se ejecutan en un ordenador con hardware de computación general o especializada (tal como un servidor web y clientes web).

El primer cliente 1, el segundo cliente 2 y la fuente 3 de datos originales están interconectados por las interfaces 4 de comunicación. Cada una de las interfaces 4 utiliza una Red de Área Local (LAN) cableada o inalámbrica, una Red de Área Metropolitana (MAN) cableada o inalámbrica o una Red de Área Ampla (WAN) cableada o inalámbrica tal como Internet o una combinación de las tecnologías de red mencionadas anteriormente y se implementan mediante cualquier protocolo de comunicación y de red adecuado.

En la Figura 2 se presenta un diagrama de secuencia de mensajes para un mecanismo según algunas realizaciones. Un sistema de base de datos distribuido comprende al menos una fuente 3 de datos originales que almacena una cantidad de datos originales. Además, el sistema de base de datos distribuido comprende una pluralidad de clientes que comprenden al menos un primer cliente 1 y un segundo cliente 2, en donde el primer cliente 1 y el segundo cliente 2 almacenan datos en la memoria caché que pueden solaparse como se describe en los casos mencionados anteriormente y que corresponden a al menos una parte de la cantidad de datos originales almacenados en la fuente de datos. El primer cliente 1 y el segundo cliente 2 tienen un modelo probabilístico que produce valores de validez asociados con los datos en la memoria caché que indican una probabilidad de que los datos en la memoria caché almacenados en el cliente correspondan a los datos originales correspondientes.

Según algunas realizaciones, en una actividad 10, el primer cliente 1 selecciona aleatoriamente consultas o submuestras de consultas de entre una pluralidad de consultas manejadas por el sistema de base de datos distribuido, en el momento en que se reciben las consultas respectivas de uno de la pluralidad de clientes. Para cada una de las consultas seleccionadas aleatoriamente, el primer cliente 1 recupera en una actividad 11 un primer fragmento de los datos en la memoria caché almacenados en el primer cliente y que corresponden a la consulta seleccionada aleatoriamente, y recupera en una actividad 12 un primer fragmento de los datos originales que corresponden a la consulta seleccionada aleatoriamente de la al menos una fuente de datos originales. Por citar un ejemplo, el primer cliente 1 puede recuperar como primera pieza de datos en la memoria caché una tabla procesable por SQL "table_A1_cached" y "table_A2_cached" ("tabla_A1_en la memoria caché" y "tabla_A2_en la memoria caché"). Además, el primer cliente 1 puede recuperar como primer fragmento de los datos originales de la fuente 3 de datos las tablas procesables por SQL "table_A1_original" y "table_A2_onginal" ("tabla_A1_original" y "tabla_A2_original").

El primer cliente 1 recupera adicionalmente en una actividad 13 para consultas que no se seleccionan aleatoriamente un segundo fragmento de los datos en la memoria caché almacenados en el primer cliente 1 y que corresponden a la consulta, en donde la consulta pertenece a las consultas que no se seleccionan aleatoriamente. Citando además el ejemplo de tablas procesables por SQL, la segunda pieza de datos en la memoria caché puede ser la tabla "table_B1_cached" ("tabla_B1_en la memoria caché"). El primer cliente 1 evalúa en una actividad 14 el valor de validez del segundo fragmento de la tabla en la memoria caché "table_B1_cached" ("tabla_B1_en la memoria caché").

El valor de validez viene dado por

$$e^{-\lambda(t-t_0)}$$

y es evaluado como valor de probabilidad de validez por el primer cliente 1, en donde t indica una tiempo actual o el tiempo estimado de recepción de los datos, por ejemplo, tabla "table_B1_cached" ("tabla_B1_en la memoria caché"). Se puede emplear una tasa λ de validez (o invalidez) para proporcionar una estimación de la probabilidad de que los datos solicitados sigan siendo válidos después de un tiempo determinado. Esto también se conoce como la probabilidad de que los datos solicitados (por ejemplo, la tabla "table_B1_cached" ("tabla_B1_en la memoria caché")) sean válidos o, en otras palabras, no estén desactualizados. La Figura 6 representa dos funciones ejemplares de esta probable precisión que disminuyen con el tiempo. La función F representa los datos solicitados que potencialmente siguen siendo más precisos (o, más correctamente, tienen una probabilidad más alta de ser válidos a lo largo del tiempo) que otros datos solicitados asociados con la función G. Por ejemplo, los datos solicitados representados por la función F tienen el 70% de probabilidad de seguir siendo válido a las 35 horas después de su última generación, mientras que los otros datos solicitados caracterizados por la función G solo son válidos hasta alrededor del 50% a las 35 horas después de su última generación. El primer cliente 1 compara el valor de probabilidad de validez para el segundo fragmento de datos en la memoria caché, por ejemplo, tabla "table_B1_cached" ("tabla_B1_en la memoria caché"), con un valor umbral dado. Si el valor de probabilidad de validez está por debajo de un umbral dado, el primer cliente 1 recupera en una actividad 15 un segundo fragmento de los datos originales que corresponden a la consulta de la al menos una fuente de datos originales, en donde el segundo fragmento de los datos originales puede ser una tabla "table_B1_original" ("tabla_B1_original"). El primer cliente 1 actualiza en una actividad 16 el segundo fragmento de los datos en la memoria caché almacenados en el primer cliente por el segundo fragmento de los datos originales, que en el caso del ejemplo descrito actualmente significa la sustitución de la tabla "table_B1_cached" ("tabla_B1_en la memoria caché") con la tabla "table_B1_original" ("tabla_B1_original").

El primer cliente 1 adapta en una actividad 17 el modelo probabilístico del primer cliente basado en el primer fragmento recuperado de los datos en la memoria caché (por ejemplo, la tabla "table_A1_cached" ("tabla_A1_en la memoria caché")) y el primer fragmento recuperado de los datos originales (por ejemplo, la tabla "table_A1_original" ("tabla_A1_original")) utilizando un algoritmo de aprendizaje automático. En particular, los datos en la memoria caché

recuperados y los datos originales recuperados se comparan para determinar si la recuperación de los datos originales habría sido apropiado o no. Es decir, si la comparación da como resultado que los datos en la memoria caché recuperados corresponden a los datos originales recuperados, la recuperación de los datos originales no habría sido conveniente ya que los datos en la memoria caché aún eran válidos; de lo contrario, habría sido sensato ya que los datos en la memoria caché no eran válidos. Este resultado que indica una eficiencia de recuperación de los datos originales de la fuente de datos originales (también conocido como sondeo) se almacena junto con los datos en la memoria caché recuperados y los datos originales recuperados, y se utiliza para adaptar el modelo probabilístico a partir del cual se deriva la tasa de validez asociada con los datos en la memoria caché. La adaptación se realiza mediante la utilización de un algoritmo de aprendizaje automático como se describe, por ejemplo, en el documento EP 3 550 447 A1. Por consiguiente, se mejora el modelo probabilístico y se adapta la decisión de si el sondeo (es decir, la recuperación de los datos originales de la base de datos y la actualización de los datos en la memoria caché) es necesario y, por lo tanto, eficiente, aumentando así la precisión del modelo de probabilidad. Esto permite reducir el número de sondeos mejorando la decisión de sondeo en comparación con el documento EP 2 908 255 A1, por ejemplo

Como un ejemplo, el primer cliente 1 puede adaptar la tasa λ de validez para el primer fragmento de los datos en la memoria caché. La tasa λ de invalidez/validez es una medida de cuánto tiempo el fragmento de los datos en la memoria caché (por ejemplo, la tabla "table_A1_cached" ("tabla_A1_en la memoria caché")) permanece válida o cómo de rápido los datos en la memoria caché se vuelven inválidos debido a cambios de los datos originales subyacentes (por ejemplo, la tabla "table_A1_original" ("tabla_A1_original")). Esta tasa de validez de un dato en la memoria caché dado se deriva, por ejemplo, estadísticamente de la ocurrencia y los resultados de (re)cálculos o (re)recopilaciones y comparaciones de los datos solicitados con su estado o valores anteriores. Por ejemplo, se ha determinado mediante una adaptación anterior del modelo probabilístico que los datos solicitados en particular tienen una tasa λ de invalidez del 10% por hora, lo que significa que la probabilidad de que los datos en la memoria caché sean válidos disminuye en un 10% cada hora. En el momento de su (re)recopilación o (re)cálculo, la tabla "table_A1_cached" ("tabla_A1_en la memoria caché") es generalmente 100% válida. Después de una hora, la tabla "table_A1_cached" ("tabla_A1_en la memoria caché") es válida con una probabilidad del 90%. Después de dos horas, la validez de la tabla "table_A1_cached" ("tabla_A1_en la memoria caché") es del 81% (= 90% disminuyó en otro 10%). Después de tres horas, la validez probable de la tabla "table_A1_cached" ("tabla_A1_en la memoria caché") es del 72,9%, y así sucesivamente.

La adaptación del modelo probabilístico llevada a cabo por el primer cliente 1 puede resultar en una nueva tasa λ de invalidez del 20% por hora para el primer fragmento de datos en la memoria caché, por ejemplo, debido a la determinación de que el primer fragmento de datos en la memoria caché se ha quedado desactualizada recientemente de forma más regular. Esto significa que después de una hora, la tabla "table_A1_cached" ("tabla_A1_en la memoria caché") es válida con una probabilidad del 80%. Después de dos horas, la validez de la tabla "table_A1_cached" ("tabla_A1_en la memoria caché") es del 64% (= 80% disminuyó en otro 20%). Después de tres horas, la validez probable de la tabla "table_A1_cached" ("tabla_A1_en la memoria caché") es del 51,2%, y así sucesivamente.

Después de la adaptación del modelo probabilístico, en una actividad 18, el primer cliente 1 pone a disposición el modelo probabilístico adaptado al segundo cliente 2. Se prevén distintas formas técnicas de puesta a disposición. Por ejemplo, el primer cliente 1 almacena los datos que especifican la adaptación del modelo probabilístico en una ubicación dada para que los recupere el segundo cliente 2 (tirar). Adicional o alternativamente, el primer cliente 1 puede enviar los datos que especifican la adaptación del modelo probabilístico al segundo cliente 2 sin ninguna solicitud de recuperación (empujar).

En algunas realizaciones, la puesta a disposición del modelo probabilístico adaptado del primer cliente 1 al segundo cliente 2 tiene en cuenta el nivel de correlación entre los datos en la memoria caché del primer cliente 1 para el que se adaptó el modelo probabilístico y los datos de la memoria caché mantenidos en el segundo cliente 2. Generalmente, las adaptaciones al modelo probabilístico del primer cliente 1 que se refieren a tales datos en la memoria caché del primer cliente 1 que tienen al menos alguna correlación significativa (por ejemplo, correlación que cumple al menos un umbral dado) con los datos en la memoria caché en el segundo cliente 2 está disponible. Las adaptaciones al modelo probabilístico del primer cliente 1 que se refieren a los datos en la memoria caché del primer cliente 1 que no tienen correlación o ninguna correlación significativa (por ejemplo, por debajo del umbral dado) pueden no estar disponibles para el segundo cliente 2 porque tales partes del modelo probabilístico del primer cliente 1, al menos actualmente, no son útiles para el segundo cliente 2.

Más específicamente, si se aplica un modelo de almacenamiento de datos según el cual todos los clientes tienen los mismos datos en la memoria caché, es decir, los datos en la memoria caché almacenados por el primer cliente 1 y almacenados por el segundo cliente 2 se refieren a los mismos datos originales subyacentes, lo que significa que los datos en la memoria caché mantenidos por el primer cliente 1 y por el segundo cliente 2 están completamente correlacionados, la actividad de hacer disponible el modelo adaptado puede ejecutarse transmitiendo cualquier tasa λ de validez actualizada para los registros de datos en la memoria caché del primer cliente 1 al segundo cliente 2. Continuando con la referencia al ejemplo anterior, el primer cliente 1 puede transmitir la tasa λ de invalidez actualizada del 20% por hora al segundo cliente 2 a través de la interfaz 4 de comunicación o enviando una indicación de que una tasa de invalidez actualizada para el primer fragmento de datos en la memoria caché ahora están disponibles en el primer cliente 1. En el segundo cliente 2, la tasa λ de validez adaptada se puede aplicar al modelo probabilístico del segundo cliente 2 para el primer fragmento de datos en la memoria caché almacenados en el segundo cliente 2. Por

tanto, el segundo cliente 2 puede adaptar su propio modelo probabilístico sustituyendo el valor actual de la tasa de invalidez para el primer fragmento de datos en la memoria caché por el valor actualizado del 20%.

Lo mismo es aplicable a los modelos de almacenamiento de datos según los cuales los datos en la memoria caché mantenidos por el primer cliente 1 se superponen parcialmente. Aquí, el valor actualizado para la tasa λ de validez del primer fragmento de datos en la memoria caché, según lo adaptado por el primer cliente 1, se pone a disposición del segundo cliente 2 si el segundo cliente 2 también almacena el primer fragmento de datos en la memoria caché y/o almacena otros fragmentos de datos en la memoria caché que están correlacionados con el primer fragmento de datos en la memoria caché. Se aplican consideraciones similares cuando se asignan tasas de validez a subconjuntos de múltiples fragmentos de datos en la memoria caché. Por ejemplo, el primer cliente 1 tiene subconjuntos A1, B1 y C1 de datos en la memoria caché, cada uno de los cuales incluye múltiples fragmentos de datos en la memoria caché que se correlacionan entre sí. Se ha establecido un modelo probabilístico para cada uno de los subconjuntos de datos en la memoria caché con las tasas λ_{A1} de validez correspondientes para el subconjunto A1 de datos en la memoria caché, λ_{B1} para el subconjunto B1 de datos en la memoria caché y λ_{C1} para el subconjunto C1 de datos en la memoria caché. El segundo cliente 2 tiene el subconjunto A2 de datos que se superpone completamente con el subconjunto A1 de datos del primer cliente 1. Además, el segundo cliente 2 tiene el subconjunto D2 de datos que se superpone parcialmente con el subconjunto C1 de datos del primer cliente 1. Además, el segundo cliente 2 tiene el subconjunto Y2 de datos que no se superpone con ningún subconjunto de datos mantenido por el primer cliente 1. El subconjunto B1 de datos mantenido por el primer cliente 1 tampoco se superpone con ningún subconjunto del segundo cliente 2.

El primer cliente 1 adapta el modelo probabilístico para los subconjuntos A1, B1 y C1 de datos, mediante el cual se obtienen tasas de validez adaptadas, a saber, λ_{A1} para el subconjunto A1 de datos, λ_{B1} para el subconjunto B1 de datos y λ_{C1} para el subconjunto C1 de datos. El primer cliente 1 y el segundo cliente 2 pueden ser conscientes de los diferentes grados de superposición entre distintos subconjuntos mantenidos por ellos mismos y los diferentes grados de correlación correspondientes entre los subconjuntos de datos en la memoria caché. Para abordar estos diferentes grados de correlación, en algunas realizaciones, el primer cliente 1 o el segundo cliente 2 pueden asociar valores de ponderación con las tasas de validez adaptadas. Para datos completamente superpuestos (alta correlación), ya que esto se puede realizar entre el subconjunto A1 de datos y el subconjunto A2 de datos, se puede asociar un valor '1' de ponderación a λ_{A1} . Para subconjuntos de datos sin superposición en absoluto (correlación baja si los fragmentos de datos en los subconjuntos son locales entre sí, o sin correlación), tal como para el subconjunto B1 de datos, se puede asignar un valor de ponderación de '0' a λ_{B1} . Para subconjuntos de datos que se superponen parcialmente (correlación media o media a alta), tal como para el subconjunto B1 de datos, se puede asignar un valor de ponderación entre cero y uno, como '0,8' a λ_{C1} . El factor de ponderación puede ser mayor, cuanto más cerca estén los datos en la memoria caché correspondientes del primer cliente 1 al segundo cliente 2 (más cerca en términos del principio de localidad mencionado anteriormente).

Utilizando los valores de ponderación, el primer cliente 1 pone, en algunas realizaciones, en la actividad 18 esas tasas de validez adaptadas disponibles, para las que se ha asignado un valor de ponderación mayor que cero. En algunas realizaciones, el primer cliente 1 pone a disposición tasas de validez adaptadas asociadas con los valores de ponderación, de modo que el segundo cliente 2 puede adaptar su propio modelo de probabilidad basándose en las tasas de validez adaptadas del primer cliente 1 y los valores de ponderación asociados. En el ejemplo específico anterior, el primer cliente 1 solo pone a disposición los valores λ_{A1} y λ_{C1} de validez, opcionalmente tuplas λ_{A1} : 1 y λ_{C1} : 0,8. El primer cliente 1 envía los parámetros λ_{A1} y λ_{C1} del modelo adaptado posiblemente acompañado con los valores de ponderación asociados al segundo cliente 2 en un mensaje que comprende una indicación de a qué subconjuntos de datos en la memoria caché pertenecen estos parámetros del modelo.

El primer cliente 1 envía una indicación al segundo cliente 2 de que los parámetros λ_{A1} y λ_{C1} del modelo adaptado ahora están disponibles en el primer cliente 1.

En un proceso de recopilación descentralizada de parámetros, diferentes clientes pueden calcular una variedad de tasas de validez según el método descrito anteriormente y mostrado en la Figura 2, en donde cada tasa λ_i de validez individual puede referirse a datos de caché (por ejemplo, conjuntos individuales de datos en la memoria caché, acciones de datos en la memoria caché correlacionados) de diferente naturaleza y campos, tales como procesamiento de imágenes, flujos de tráfico, etc. En una entidad central que actúa como un servidor de parámetro, como un cliente específico o una unidad de cálculo masiva, estas tasas λ_i de validez pueden recopilarse y procesarse posteriormente, por ejemplo, para evaluar el proceso de aprendizaje de los clientes individuales, para monitorizar el funcionamiento sin fallos de los clientes, etc.

El intercambio de parámetros del modelo como, por ejemplo, las tasas λ de validez en un aprendizaje automático distribuido (por ejemplo, entre iguales) alivia una entidad central que tradicionalmente mantiene el modelo probabilístico para rastrear las probabilidades de invalidez de los datos en la memoria caché (como, por ejemplo, se describe en la solicitud de Patente Europea número 19 166 538 y los otros documentos de la técnica anterior mencionados anteriormente) de la red y la carga de cálculo, y así elimina un problema técnico de cuello de botella en tal sistema. Dado que la capacidad de aprendizaje automático llevada a cabo por el primer cliente 1 puede ser reutilizada por otro cliente, se ofrece la posibilidad de adaptar constantemente el modelo probabilístico, lo que hace que el proceso de aprendizaje automático sea más efectivo y aumenta la resiliencia general de toda la red.

En algunas realizaciones mostradas en la Figura 3, el primer cliente 1 recupera para consultas seleccionadas aleatoriamente un primer fragmento adicional de datos en la memoria caché local almacenados por al menos otro cliente de la pluralidad de clientes y que corresponden a la consulta seleccionada aleatoriamente (actividad 11a). Alternativa o adicionalmente, el primer cliente 1 puede adquirir en una actividad 11a datos adicionales en la memoria caché en una fuente de datos adicional. Por un lado, estos fragmentos adicionales de datos en la memoria caché se pueden utilizar para preparar y devolver la respuesta a la consulta al cliente solicitante. Por otro lado, estos fragmentos adicionales de datos en la memoria caché se pueden utilizar para adaptar de manera más universal el modelo probabilístico del primer cliente 1 y, por lo tanto, para acelerar el proceso de aprendizaje automático en el primer cliente 1.

Con este fin, el primer cliente 1 también puede recuperar, para cada fragmento adicional de datos en la memoria caché, un fragmento adicional de los datos originales que correspondan a la consulta seleccionada aleatoriamente de la al menos una fuente de datos originales. A medida que uno o más fragmentos adicionales de datos corresponden a los criterios de búsqueda de la consulta, se puede suponer al menos un cierto nivel de correlación entre el primer fragmento de datos y los uno o más fragmentos de datos adicionales. Por lo tanto, se puede utilizar una determinación de la validez de uno o más fragmentos de datos adicionales (por ejemplo, mediante la comparación de los fragmentos de datos adicionales en la memoria caché con los fragmentos de datos originales correspondientes como se describe anteriormente para el primer fragmento de datos) para adaptar el modelo probabilístico del primer cliente de forma más eficaz, utilizando una base más amplia de datos de aprendizaje. Los valores de ponderación que indican un nivel de correlación entre fragmentos adicionales de datos en la memoria caché y el primer fragmento de datos en la memoria caché, así como los valores de validez actuales de los fragmentos adicionales de datos en la memoria caché (por ejemplo, proporcionados por la fórmula mencionada anteriormente) pueden recuperarse del otro cliente u otra fuente de datos y utilizarse para adaptar el modelo probabilístico del primer cliente. Opcionalmente, el primer cliente 1 puede incluir uno o más fragmentos adicionales de datos en la memoria caché recuperados (posiblemente actualizados por los fragmentos recuperados de datos originales) en los datos en la memoria caché almacenados en el primer cliente 1.

Como ejemplo, el cliente 1 puede recuperar la tabla "table_a2_cached" ("tabla_a2_en la memoria caché") del segundo cliente 2 y utilizarla junto con la tabla "table_A1_cached" (tabla_A1_en la memoria caché") para la adaptación del modelo probabilístico. Dado que en un entorno de aprendizaje automático distribuido implementado en una red entre iguales, cada cliente puede realizar simultáneamente las funciones del primer cliente 1, el tráfico de red que se produce cuando, por ejemplo, las tablas se transmiten a los clientes que actúan como primeros clientes ya no se dirigen a un solo nodo, sino que se dirigen a más fuentes. Por lo tanto, el tráfico de la red puede distribuirse de manera más uniforme en el entorno informático distribuido.

Con referencia ahora a la Figura 4, en algunas realizaciones, después de la adaptación del modelo probabilístico por el primer cliente 1, el segundo cliente 2 recupera en una actividad 19 al menos una parte del modelo probabilístico adaptado del primer cliente 1 y además adapta en una actividad 20 su propio modelo probabilístico basado en el modelo probabilístico adaptado recuperado del primer cliente 1. Nuevamente, tomando el ejemplo antes mencionado de los subconjuntos C1 y D2 de datos parcialmente superpuestos, que están en manos del primer cliente 1 y del segundo cliente 2 respectivamente, el segundo cliente 2 recibe la tasa λ'_{C1} de validez ya adaptada por el primer cliente 1 basada en el subconjunto C1 de datos, por ejemplo, en la tabla "table_A1_cached" (tabla_a1_en la memoria caché") y calcula una tasa λ'_{D2} de validez adicional adaptada que puede basarse en el subconjunto D2 de datos, por ejemplo, la tabla "table_A2_cached" ("tabla_A2_en la memoria caché").

Como se mencionó anteriormente, en algunas realizaciones, el segundo cliente 2 recibe una tasa de validez actualizada para los primeros datos en la memoria caché asociados con un valor de ponderación. El valor de ponderación indica un nivel de correlación del primer fragmento de datos en la memoria caché almacenados en el primer cliente 1 con otro fragmento de datos en la memoria caché almacenados en el segundo cliente 2. En tales realizaciones, adaptar el modelo probabilístico del segundo cliente 2 basándose en el modelo probabilístico adaptado recuperado del primer cliente 1 comprende adaptar el modelo probabilístico del segundo cliente 2 basándose en el valor de validez actualizado para los primeros datos en la memoria caché asociados y el valor de ponderación asociado. Más específicamente, el segundo cliente puede aplicar la tasa de validez actualizada recibida del primer cliente 1 en una tasa de validez de su propio modelo probabilístico que indica una probabilidad de validez del otro fragmento de datos en la memoria caché del segundo cliente 2, estando correlacionada el otro fragmento de datos en la memoria caché del segundo cliente 2 con el primer fragmento de datos en la memoria caché del primer cliente 1 en un grado indicado por el valor de ponderación asociado.

Si el valor de ponderación indica una correlación completa (por ejemplo, valor 1 de ponderación), el segundo cliente puede incluir la tasa de validez actualizada recibida del primer cliente 1 en su propio modelo probabilístico, es decir, la tasa de validez de los otros datos de caché del segundo cliente 2, sin descuento, es decir, de manera similar a las actualizaciones nativas obtenidas mediante la ejecución análoga del mecanismo de la Figura 2 en el segundo cliente 2. Por ejemplo, en caso de una correlación completa, el segundo cliente 2 puede sustituir la tasa de validez actual de los otros datos en la memoria caché por la tasa de validez actualizada recibida del primer cliente 1. El segundo cliente 2 también puede actualizar la tasa de validez de los otros datos en la memoria caché incluyendo la tasa de validez actualizada recibida del primer fragmento de datos en la memoria caché del primer cliente 1 de otras formas, por ejemplo mediante suavizado exponencial.

Si el valor de ponderación indica un grado más bajo de correlación (por ejemplo, el valor de ponderación 0,8 indica una correlación del 80%), el segundo cliente 2 puede integrar la tasa de validez actualizada recibida del primer cliente 1 de una manera descontada. Por ejemplo, el segundo cliente puede calcular un promedio ponderado actualizado utilizando o añadiendo la tasa de validez actualizada recibida del primer cliente a la serie de tasas de validez de un suavizado exponencial de una manera descontada, por ejemplo, multiplicando la tasa de validez actualizada recibida del primer cliente 0,8 con el valor de ponderación asociado.

Además, el segundo cliente 2 también puede adaptar las tasas de validez de otros fragmentos de datos en la memoria caché almacenados en el segundo cliente 2 que están correlacionados con el otro fragmento de datos en la memoria caché a la que se refiere el valor de ponderación recibido. A este respecto, el segundo cliente 2 puede derivar valores de ponderación adicionales dependiendo del nivel de correlación entre el otro fragmento de datos en la memoria caché y los otros fragmentos de datos en la memoria caché. Por ejemplo, un fragmento adicional de los datos en la memoria caché puede correlacionarse con el otro fragmento de los datos en la memoria caché en un grado del 90%, lo que da como resultado un valor de ponderación de $0,8 \times 0,9 = 0,72$. Por lo tanto, la tasa de validez actualizada recibida del primer cliente 1 se puede aplicar para actualizar la tasa de validez de este fragmento adicional de datos en la memoria caché del segundo cliente utilizando un valor de ponderación de 0,72, que refleja la correlación entre el primer fragmento de datos en la memoria caché de el primer cliente 1 y el otro fragmento de los datos en la memoria caché del segundo cliente 2 por un lado y el otro fragmento de los datos en la memoria caché y el fragmento adicional de los datos en la memoria caché del segundo cliente 2 por otro lado.

Con referencia a la Figura 5, antes de la adaptación de su propio modelo probabilístico, el primer cliente 1 recupera en una actividad 17a del segundo cliente 2 al menos una parte del modelo probabilístico adaptado del segundo cliente 2. Tomando nuevamente el ejemplo antes mencionado de los subconjuntos C1 y D2 de datos parcialmente superpuestos, el primer cliente 1 puede recuperar una tasa λ^{D2} de validez del segundo cliente 2 en el que λ^{D2} puede haber sido adaptado en la tabla "table_A2_cached" ("tabla_A2_en la memoria caché") y posteriormente adapta en la actividad 17 su propio modelo probabilístico basado en el modelo probabilístico adaptado recuperado (λ^{D2}) del segundo cliente 2 y además en el subconjunto C1 de datos, por ejemplo, tabla "table_A1_cached" (tabla_A1_en la memoria caché). Por lo tanto, el primer cliente 1 obtendría una tasa λ^{C1} de validez. En la actividad 20, el primer cliente 1 elabora su modelo probabilístico adaptado, tal como la tasa λ^{C1} de validez adaptada, disponible para otros clientes.

En la descripción del tema, se puede establecer cualquier algoritmo de aprendizaje automático que se pueda utilizar en conexión con modelos probabilísticos. Por ejemplo, el algoritmo de aprendizaje automático, dada una pluralidad de consultas de búsqueda seleccionadas convertidas en un conjunto de características estructuradas, construye un modelo probabilístico para predecir un objetivo (por ejemplo, primer valor de indicador almacenado con las consultas de búsqueda). El término "característica" aquí se refiere a cualquier información que pueda deducirse de las consultas de búsqueda. La conversión de la pluralidad de consultas de búsqueda seleccionadas puede comprender calcular valores de todas las características, utilizando los criterios de búsqueda y/o datos de referencia (por ejemplo, tablas que muestran valores de salida predeterminados para el circuito electrónico dado, etc.). Por ejemplo, a partir de todos los criterios de búsqueda de la pluralidad de consultas de búsqueda y los datos de referencia, se puede seleccionar el conjunto de características que utilizará el algoritmo de aprendizaje automático. Por lo tanto, el conjunto de características puede contener una mezcla de criterios de búsqueda y características que requieren cálculos, pero una característica también puede corresponder a uno o más criterios de búsqueda (por ejemplo, todos los parámetros de entrada para un circuito electrónico dado para ser simulados por las simulaciones de circuito electrónico basadas en ordenador, marca de tiempo de la consulta de búsqueda, etc.). El conjunto de características que se estructura como el algoritmo de aprendizaje automático requiere la entrada de, por ejemplo, un archivo de datos estructurados (por ejemplo, un archivo de datos separados por comas), con una fila por consulta de búsqueda y una columna por característica. El orden de las columnas sigue un patrón determinado para que el modelo probabilístico se utilice correctamente una vez implementado. El algoritmo de aprendizaje automático también puede construir iterativamente un árbol de decisión. A partir de la pluralidad de consultas de búsqueda seleccionadas, el algoritmo de aprendizaje automático prueba características y condiciones de estas características para dividir la pluralidad en dos conjuntos secundarios. Una de estas pruebas puede ser "si los parámetros de entrada para el circuito electrónico dado son iguales a una tensión que tiene un valor de 10" (es decir, en general, verificar si la característica es igual a un valor dado), o "es el parámetro de entrada para el circuito electrónico dado igual a una tensión que tiene un valor superior a 20" (es decir, en general, verificar si la característica es mayor/menor que el valor dado). De todas las características y condiciones probadas, el algoritmo de aprendizaje automático conserva solo la característica y condición que separa la mayoría de la pluralidad en un primer conjunto secundario que contiene consultas de búsqueda con su objetivo igual a un valor, y un segundo conjunto secundario que contiene las consultas de búsqueda con su objetivo que no es igual al valor uno. En otras palabras, el árbol de decisión lo construye el algoritmo de aprendizaje automático de tal manera que una característica se toma como un nodo con dos trayectorias de salida, una para consultas de búsqueda que tiene la característica que es igual al valor dado (el primer conjunto secundario) y otra trayectoria para consultas de búsqueda que tienen la característica que no es igual al valor dado (el segundo conjunto secundario). Es decir, las consultas de búsqueda se dividen en dos conjuntos secundarios según la función. El algoritmo de aprendizaje automático construye progresivamente el árbol de decisión, y luego también divide los conjuntos secundarios en conjuntos más pequeños con la misma lógica de selección de características y condiciones. El algoritmo de aprendizaje automático se detiene una vez que el árbol de decisión alcanza un tamaño o complejidad determinados proporcionados en el parámetro del mismo.

La Figura 7 es una representación esquemática del componente interno de un ordenador del primer cliente 1, del segundo cliente 2 y/o de la fuente 3 de datos originales. El ordenador 100 incluye un conjunto de instrucciones para hacer que el ordenador 100 realice cualquiera de las metodologías tratadas en la presente memoria cuando se ejecuta por el ordenador 100. El ordenador 100 incluye al menos un procesador 101, una memoria principal 106 y un dispositivo 103 de interfaz de red que se comunican entre sí a través de un bus 104. Opcionalmente, el ordenador 100 puede incluir además una memoria estática 105 y una unidad de disco. Se pueden proporcionar una pantalla de video, un dispositivo de entrada alfanumérico y un dispositivo de control del cursor como ejemplos de la interfaz 102 de usuario. El dispositivo 103 de interfaz de red conecta el ordenador 100 a los otros componentes del sistema informático distribuido tal como el primer cliente 1, el segundo cliente 2, la fuente 3 de datos originales o los componentes adicionales.

El ordenador 100 también aloja la memoria caché 107. La memoria caché 107 puede almacenar las tablas de base de datos recibidas también en una memoria caché. La memoria caché 107 puede estar compuesta por componentes de hardware y de software que almacenan las tablas de la base de datos de modo que las solicitudes futuras de las tablas de la base de datos puedan ser atendidas más rápidamente que sin almacenamiento en la memoria caché. Puede haber memorias caché basadas en hardware, tales como memorias caché de CPU, memorias caché de GPU, procesadores de señales digitales y búferes de traducción anticipada, así como memorias caché basadas en software tales como memorias caché de páginas, memorias caché de web (Protocolo de Transferencia de Hipertexto, HTTP, memorias caché), etc.

Un conjunto de instrucciones ejecutables por ordenador (es decir, código de programa informático) que incorpora una cualquiera o todas las metodologías descritas en la presente memoria, reside completamente, o al menos parcialmente, en un medio legible por ordenador, por ejemplo, la memoria principal 106. La memoria principal 106 aloja el código del programa informático para entidades funcionales tales como el procesamiento 108 de solicitudes de base de datos que incluye la funcionalidad para recibir y procesar solicitudes de base de datos y la funcionalidad 109 de procesamiento de datos. Las instrucciones pueden transmitirse o recibirse además como una señal propagada vía Internet a través del dispositivo 103 de interfaz de red o a través de la interfaz 102 de usuario. La comunicación dentro del ordenador se realiza a través del bus 104. El funcionamiento básico del ordenador 100 está controlado por un sistema operativo que también se encuentra en la memoria principal 106, el al menos un procesador 101 y/o la memoria estática 105.

En general, las rutinas ejecutadas para implementar las realizaciones, ya sean implementadas como parte de un sistema operativo o una aplicación, componente, programa, objeto, módulo o secuencia de instrucciones específicos, o incluso un subconjunto de los mismos, pueden denominarse en la presente memoria "código de programa informático" o simplemente "código de programa". El código de programa generalmente comprende instrucciones legibles por ordenador que residen en distintos momentos en distintos dispositivos de memoria y almacenamiento en un ordenador y que, cuando son leídas y ejecutadas por uno o más procesadores en un ordenador, hacen que el ordenador realice las operaciones necesarias para ejecutar las operaciones y/o elementos que incorporan los distintos aspectos de las realizaciones de la invención. Las instrucciones de programa legibles por ordenador para llevar a cabo operaciones de las realizaciones de la invención pueden ser, por ejemplo, lenguaje ensamblador o código fuente o código objeto escrito en cualquier combinación de uno o más lenguajes de programación.

REIVINDICACIONES

1. Un método para procesar consultas en un sistema de base de datos distribuido, comprendiendo el sistema de base de datos distribuido:

al menos una fuente de datos originales que almacena una cantidad de datos originales, y

5 una pluralidad de clientes que comprenden al menos un primer cliente y un segundo cliente, en donde el primer cliente y el segundo cliente almacenan datos en la memoria caché que corresponden a al menos una parte de la cantidad de datos originales;

en donde el primer cliente y el segundo cliente alojan un modelo probabilístico que produce valores de validez asociados con los datos en la memoria caché que indican una probabilidad de que los datos en la memoria caché almacenados en el cliente coincidan con los datos originales correspondientes;

10

caracterizado por que el método comprende, en el primer cliente:

seleccionar aleatoriamente consultas de entre una pluralidad de consultas manejadas por el sistema de base de datos distribuido, en el momento en que se recibe la consulta respectiva de uno de la pluralidad de clientes;

para cada una de las consultas seleccionadas aleatoriamente,

15 recuperar un primer fragmento de los datos en la memoria caché almacenados en el primer cliente y que corresponden a la consulta seleccionada aleatoriamente, y

recuperar un primer fragmento de los datos originales que corresponden a la consulta seleccionada aleatoriamente de la al menos una fuente de datos originales;

para las consultas que no se seleccionan aleatoriamente,

20 recuperar un segundo fragmento de los datos en la memoria caché almacenados en el primer cliente y que corresponden a la consulta;

evaluar el valor de validez del segundo fragmento de los datos en la memoria caché almacenados en el primer cliente;

si el valor de validez está por debajo de un umbral determinado,

25 recuperar un segundo fragmento de los datos originales que corresponden a la consulta de la al menos una fuente de datos originales, y

actualizar el segundo fragmento de los datos en la memoria caché almacenados en el primer cliente mediante el segundo fragmento de los datos originales; y

30 adaptar el modelo probabilístico del primer cliente basándose en el primer fragmento recuperado de los datos en la memoria caché y el primer fragmento recuperado de los datos originales utilizando un algoritmo de aprendizaje automático,

poner a disposición del segundo cliente el modelo probabilístico adaptado.

35 2. El método según la reivindicación 1, que comprende además, para consultas seleccionadas aleatoriamente, recuperar un primer fragmento adicional de datos en la memoria caché almacenados por al menos un cliente de la pluralidad de clientes y que corresponden a la consulta seleccionada aleatoriamente.

3. El método según la reivindicación 2, en donde la adaptación del modelo probabilístico se basa además en el primer fragmento adicional de datos en la memoria caché.

4. El método según una cualquiera de las reivindicaciones 1 a 3, que además comprende, en el segundo cliente:

recibir al menos una parte del modelo probabilístico adaptado del primer cliente procedente del primer cliente;

40 adaptar el modelo probabilístico del segundo cliente basándose en el modelo probabilístico adaptado recuperado del primer cliente.

45 5. El método según la reivindicación 4, en donde poner a disposición del segundo cliente el modelo probabilístico adaptado comprende poner a disposición del segundo cliente un valor de validez actualizado para los primeros datos en la memoria caché asociados con un valor de ponderación, en donde el valor de ponderación indica un nivel de correlación del primer fragmento de datos en la memoria caché almacenados en el primer cliente con otro fragmento de datos en la memoria caché almacenados en el segundo cliente, y en donde adaptar el modelo probabilístico del segundo cliente basándose en el modelo probabilístico adaptado recuperado del primer cliente comprende adaptar el

modelo probabilístico del segundo cliente basándose en el valor de validez actualizado para los primeros datos en la memoria caché asociados y el valor de ponderación asociado.

6. El método según cualquiera de las reivindicaciones 1 a 5, que además comprende, en el primer cliente:

recuperar al menos una parte del modelo probabilístico adaptado del segundo cliente procedente del segundo cliente;

5 en donde la adaptación del modelo probabilístico del primer cliente se basa además en el modelo probabilístico adaptado recuperado del segundo cliente.

7. Un ordenador que actúa como un primer cliente para manejar datos en un entorno informático distribuido que comprende una pluralidad de clientes que comprenden al menos un primer cliente y un segundo cliente y al menos una fuente de datos originales que almacena una cantidad de datos originales, estando dispuesto el ordenador para ejecutar el método de cualquiera de las reivindicaciones 1 a 6.

10 8. Un producto de programa informático que comprende instrucciones de código de programa almacenadas en un medio legible por ordenador para ejecutar las etapas del método según cualquiera de las reivindicaciones 1 a 6 cuando dicho programa ejecuta en un ordenador.

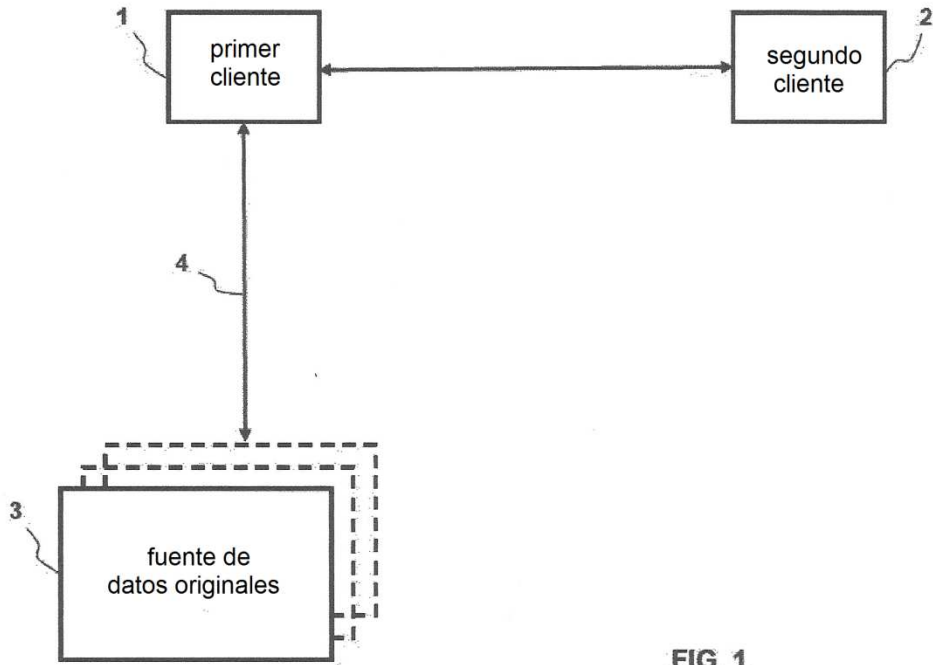


FIG. 1

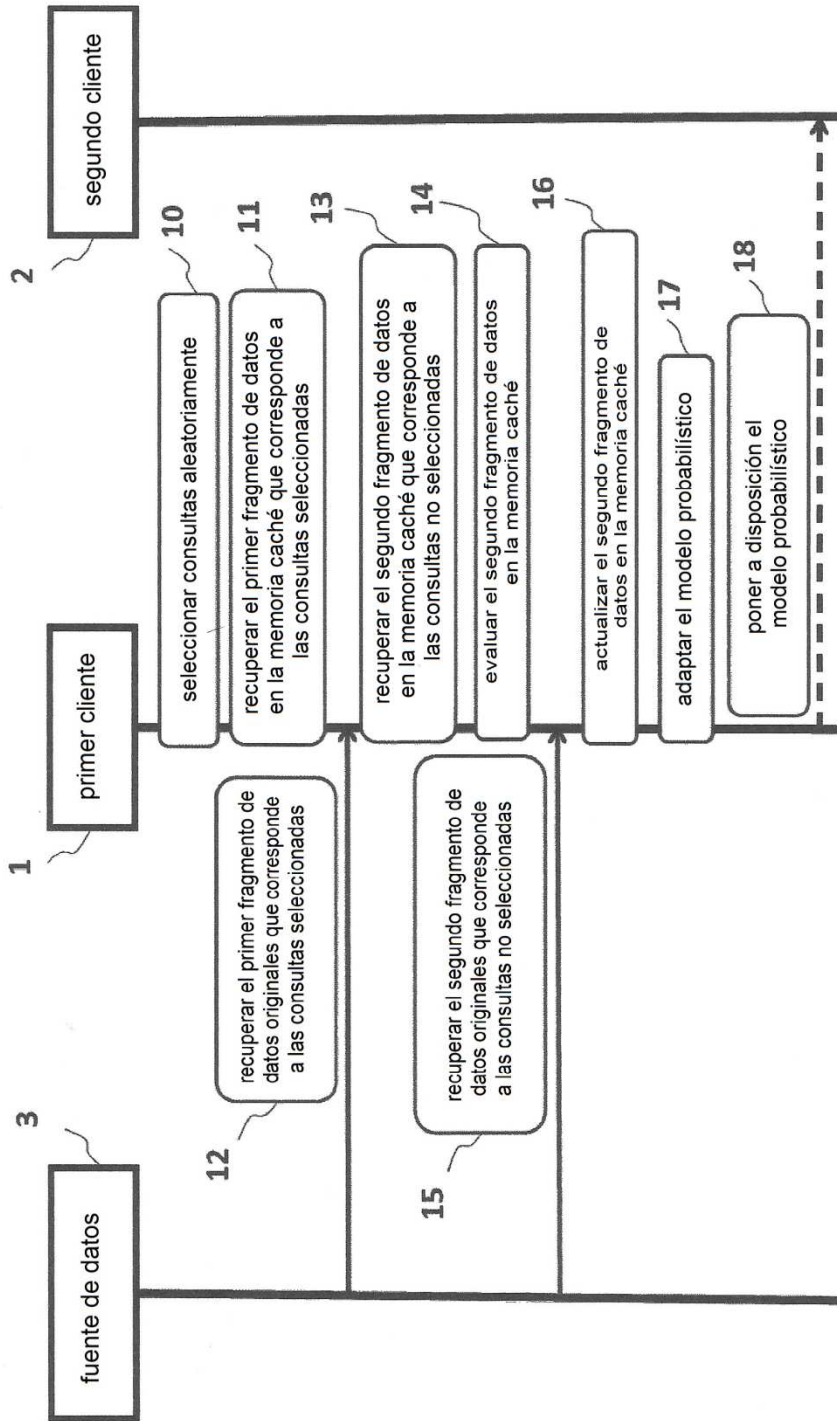


FIG. 2

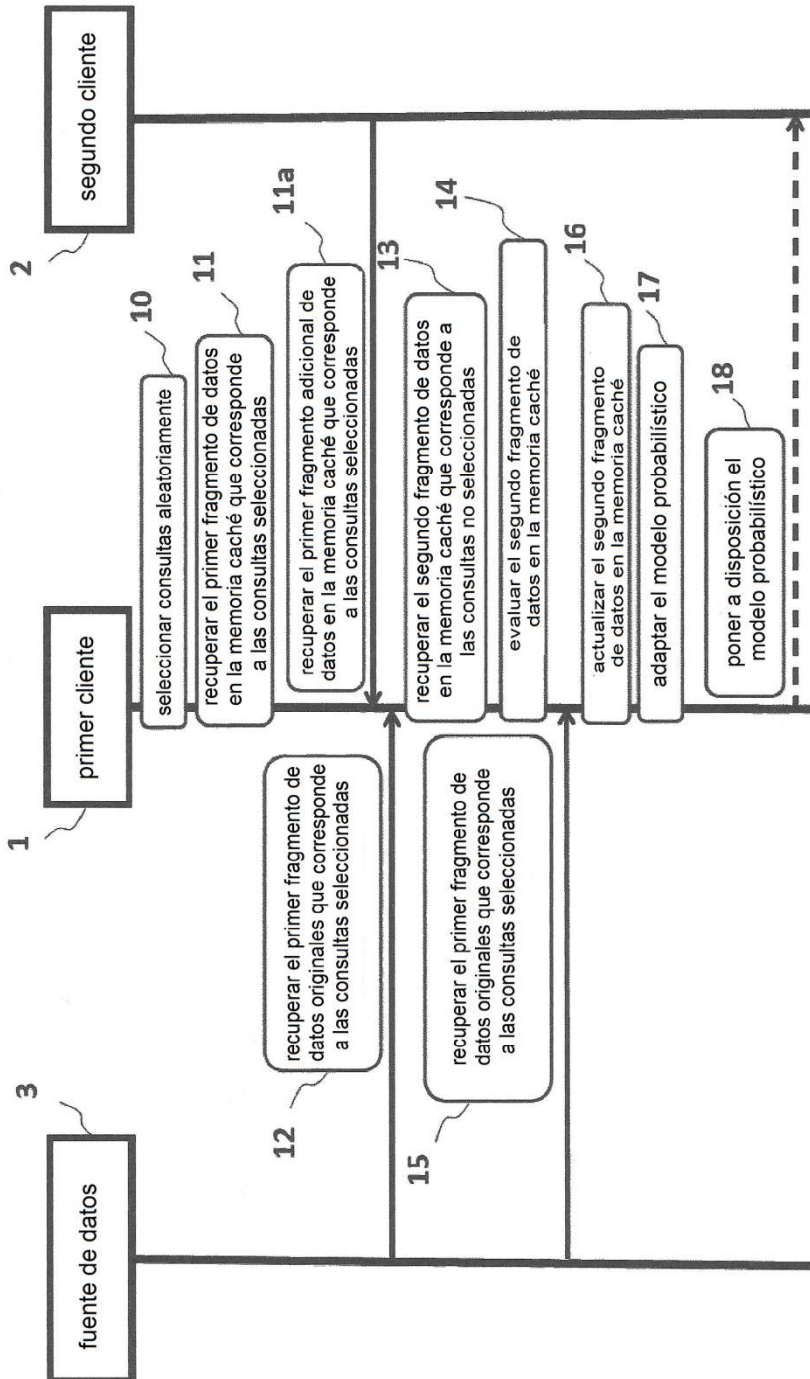


FIG. 3

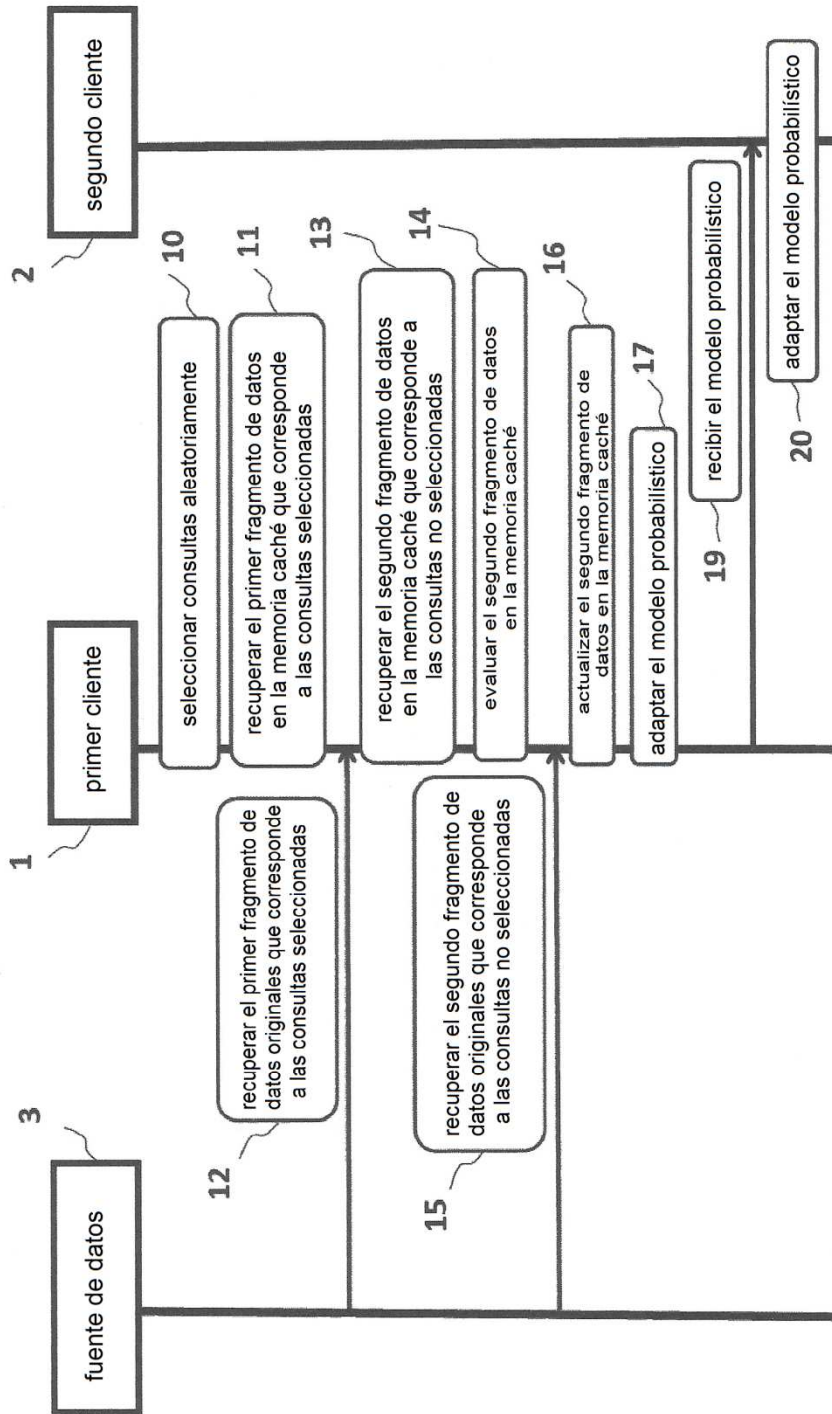


FIG. 4

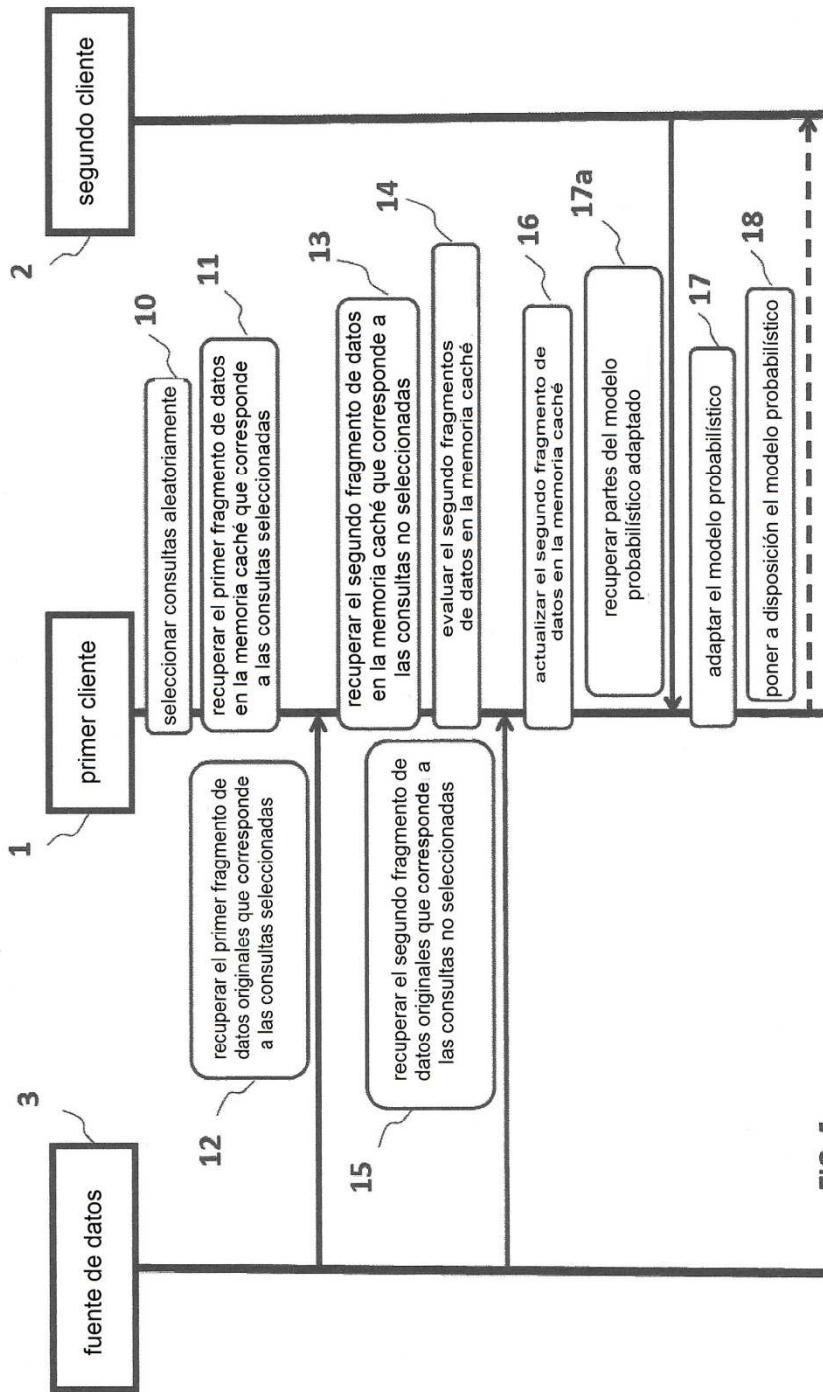


FIG. 5

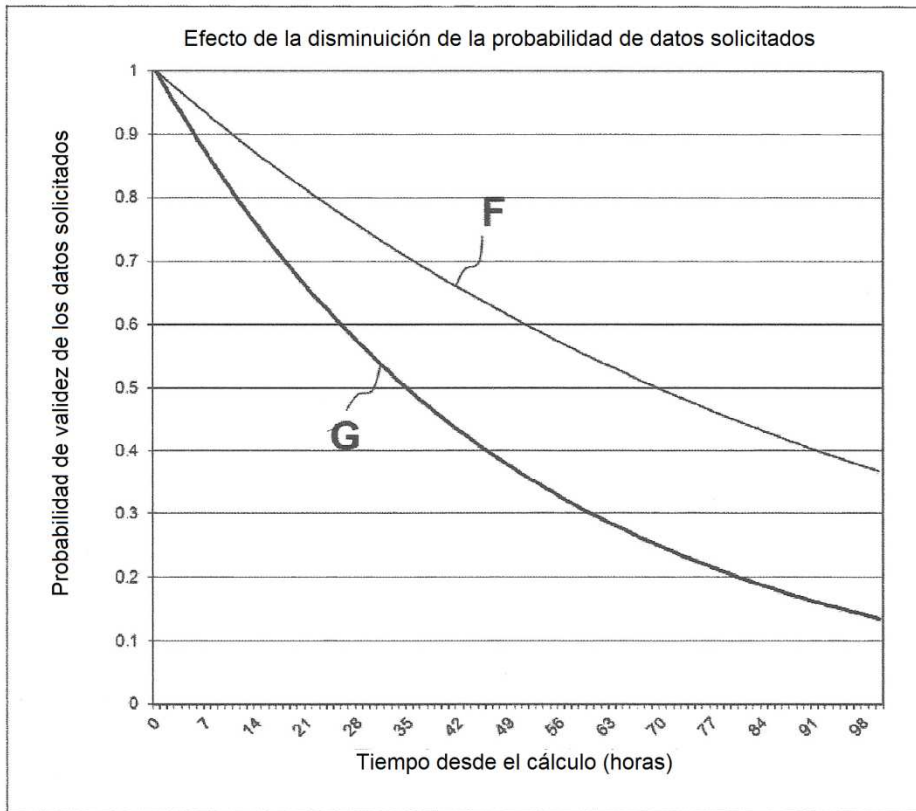


FIG. 6

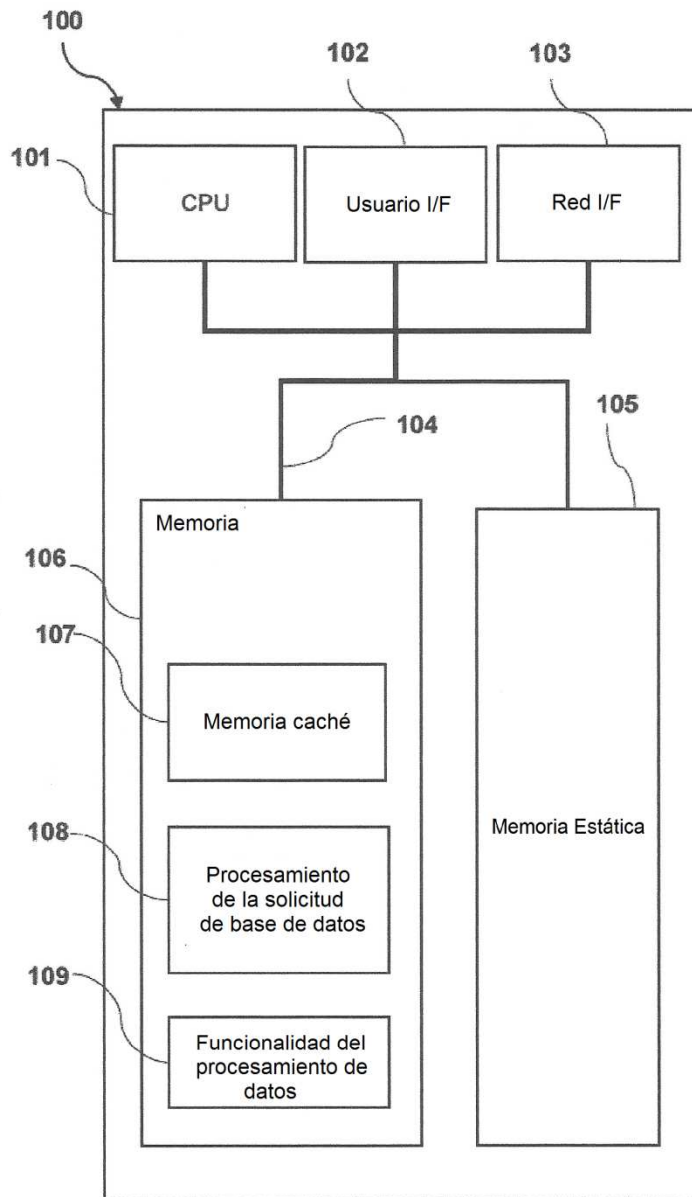


FIG. 7