

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 609 445**

51 Int. Cl.:

G06F 17/30 (2006.01)

G06N 5/02 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **28.06.2012 E 12174230 (8)**

97 Fecha y número de publicación de la concesión europea: **14.12.2016 EP 2674875**

54 Título: **Método, controlador, programa y sistema de almacenamiento de datos para realizar procesado de reconciliación**

30 Prioridad:

11.06.2012 GB 201210234

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

20.04.2017

73 Titular/es:

**FUJITSU LIMITED (100.0%)
1-1, Kamikodanaka 4-chome, Nakahara-ku
Kawasaki-shi, Kanagawa 211-8588, JP**

72 Inventor/es:

**HU, BO y
LAERA, LOREDANA**

74 Agente/Representante:

UNGRÍA LÓPEZ, Javier

ES 2 609 445 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

Método, controlador, programa y sistema de almacenamiento de datos para realizar procesado de reconciliación

5 La presente invención se refiere al campo del almacenamiento de datos y el procesado asociado. Específicamente, realizaciones de la presente invención se refieren al rendimiento de recursos de procesado de reconciliación en una representación gráfica de un conjunto de datos. El procesado de reconciliación tiene la finalidad de reconciliar la heterogeneidad entre recursos semánticamente correspondientes en el gráfico.

10 El enorme volumen de datos gráficos disponibles crea un potencial de análisis automatizado o semiautomatizado que puede no solamente poner de manifiesto tendencias estadísticas, sino también descubrir configuraciones ocultas y obtener conocimiento de datos. La semántica formal desempeña un papel clave en la automatización de tareas que requieren mucho cálculo. Aunque hay una larga discusión sobre cómo se captura mejor la semántica, se considera ampliamente que los gráficos y las representaciones a modo de gráficos son el mejor instrumento para emular cómo los humanos perciben el mundo (como una ontología con entidades y relaciones entre entidades).

15 Por lo tanto, las bases de datos gráficos ofrecen la ventaja de presentar naturalmente una representación de conocimientos basada en “redes semánticas” que puede almacenar grandes cantidades de datos estructurados y no estructurados.

20 Una base de datos gráficos es una representación de datos que emplea nodos y bordes (o arcos) para representar entidades, y arcos entre nodos para representar relaciones entre dichas entidades. Se usan bases de datos gráficos en una amplia variedad de diferentes aplicaciones que en general se pueden agrupar en dos categorías principales. La primera consta de sistemas complejos basados en conocimiento que tienen grandes colecciones de descripciones de conceptos (denominadas “aplicaciones basadas en conocimiento”), tal como soporte de decisiones inteligentes y autoaprendizaje. La segunda incluye aplicaciones que implican realizar análisis de gráficos sobre datos transaccionales (denominadas “aplicaciones de datos transaccionales”), tal como inteligencia de empresas y datos sociales.

25 En el corazón de las bases de datos gráficos formalizadas está el Marco de Descripción de Recursos, RDF, un simple lenguaje de modelado de datos basado en gráficos que realiza marcación semántica de datos. Con RDF, los silos de datos pueden empezar a fragmentarse conjuntamente y el paisaje de datos dispersos actuales transformarse en un gráfico de datos conectado sobre el que se puede construir complicadas aplicaciones de analítica de datos e inteligencia comercial.

30 Los conjuntos de datos pueden ser en general altamente heterogéneos y distribuidos. La naturaleza descentralizada de tales datos origina el problema de que a menudo muchas fuentes de datos usan diferentes referencias para indicar el mismo objeto del mundo real. Un paso necesario e importante hacia la utilización de datos gráficos disponibles es efectivamente identificar y reconciliar múltiples referencias para consistencia semántica. A continuación, el término “reconciliación” se usa para indicar el proceso de reconciliar la heterogeneidad entre recursos (como nodos en un gráfico de datos, por ejemplo, como el sujeto u objeto de triplas RDF) identificando y definiendo enlaces de equivalencia entre recursos que corresponden semánticamente uno a otro. Se sigue que el “procesado de reconciliación” es la ejecución de algoritmos e instrucciones con un procesador con el fin de lograr la reconciliación.

35 El significado de reconciliación de datos es evidente. La reconciliación de datos asegura la integridad de los datos cuando se enlazan conjuntos de datos heterogéneos (dando lugar a variedad semántica en los datos). El análisis significativo no puede ser realizado de otro modo. Mientras tanto, las equivalencias permiten alinear aplicaciones una con otra. Por lo tanto, las comunicaciones entre las aplicaciones se pueden automatizar y delegar a los ordenadores.

40 NATALYA F NOY y colaboradores: “The PROMPT suite: Interactive tools for ontology merging and mapping”, INTERNATIONAL JOURNAL OF HUMAN COMPUTER STUDIES, vol. 59, nº 6, 1 Diciembre 2003 (2003-12-01), páginas 983-1024, XP055055662, ISSN: 1071-5819, DOI: 10,1016/j.ijhcs.2003,08,002, describe una herramienta que permite al usuario seleccionar secciones de dos ontologías a fusionar en un tiempo determinado por el usuario.

45 Las realizaciones de la presente invención proporcionan un método para realizar reconciliación de recursos representados en un gráfico de datos como nodos; incluyendo el método: seleccionar un subconjunto inicial de los nodos del gráfico para los que se ha de realizar procesado de reconciliación con un nodo particular; realizar procesado de reconciliación entre el nodo particular y cada uno del subconjunto inicial de nodos; retardar el procesado de reconciliación entre el nodo particular y otro subconjunto de los nodos del gráfico para el que se ha de realizar reconciliación con el nodo particular, donde el procesado de reconciliación del subconjunto adicional de los nodos se retarda y realiza durante intervalos de tiempo cuando la supervisión de recursos informáticos responsables del procesado de reconciliación indica que dichos recursos informáticos estarían en reposo en caso contrario.

50 Se puede crear enlaces de equivalencia durante o después de la creación de un gráfico. En procesos de

reconciliación existentes, el proceso de hallar enlaces de equivalencia se realiza simultáneamente (en una sola rutina) frente a todo el conjunto de datos. Esto podría ser una fuente significativa de ineficiencia si se considera una gran cantidad de datos (del tamaño de miles de millones de nodos). La reconciliación es un proceso complejo, tedioso y lento, así como generalmente caro. Los algoritmos de reconciliación realizan a menudo operaciones computacionalmente intensivas que pueden ser considerablemente lentas. Esto plantea varios retos con vistas a su aplicabilidad práctica para tareas complejas reales y la eficiente utilización de las arquitecturas de computación que mejor encajen en los requisitos de datos para obtener máximo rendimiento y escalabilidad. La cantidad de datos disponibles está creciendo y, por lo tanto, la cantidad de datos disponibles a reconciliar a menudo no puede ser manejada por los acercamientos y las herramientas existentes. Para poder reconciliar grandes cantidades de datos, es deseable un enfoque severo en cuestiones de escalabilidad.

Los datos del gráfico pueden denominarse “datos conectados”, “datos de gráfico”, “datos enlazados” o “datos relacionados”, entre otras expresiones –todas las cuales pretenden reflejar la estructura conceptual de un gráfico como una pluralidad de nodos interconectados por arcos. En algunas implementaciones, los datos del gráfico pueden ser “datos enlazados” como en datos proporcionados como parte de la iniciativa “Datos abiertos enlazados” (LOD), aunque las realizaciones de la presente invención no se limitan a tales implementaciones, y el término “datos enlazados” puede ser interpretado en sentido más amplio que solamente datos proporcionados como parte de la iniciativa LOD.

Proporcionando un proceso con el que se realiza el procesado de reconciliación, pero no necesariamente todo en una sola rutina, se habilita la escalabilidad. Es decir, cierto procesado de reconciliación puede ser realizado a la adición de un nuevo nodo al gráfico, o a la adición de un nuevo elemento de datos (tal como una tripla) que tiene un nodo como sujeto, pero el procesado de reconciliación no es necesariamente con todo el conjunto de nodos en el gráfico, simplemente un subconjunto. El tamaño de dicho subconjunto inicial no tiene que aumentar necesariamente a medida que el tamaño del gráfico aumenta, y por lo tanto el procesado requerido para el procesado de reconciliación inicial no es inviable a medida que el gráfico aumenta.

Además, reduciendo el requisito de realizar procesado de reconciliación frente a todo el gráfico en una pasada o una rutina, el cuello de botella producido por el procesado de reconciliación de nuevos nodos se alivia. Esto mejora el rendimiento del sistema en conjunto, y también permite al operador del sistema tender hacia algoritmos de reconciliación más complejos y por lo tanto potencialmente más efectivos, sin preocuparse por el impacto adverso en el rendimiento del sistema.

La reconciliación de recursos puede ser un proceso de reconciliar la heterogeneidad entre recursos en un gráfico identificando y produciendo enlaces de equivalencia entre recursos (representados por nodos en el gráfico) que corresponden uno a otro semánticamente. Por ejemplo, donde dos recursos que tienen nombres diferentes se refieren al mismo objeto del mundo real (es decir, los dos recursos son semánticamente equivalentes/de significado equivalente), sería apropiado añadir un enlace al gráfico indicando que los dos recursos son equivalentes. Reconciliar recursos puede incluir identificar dónde existen equivalencias entre recursos (los nodos de gráfico son un recurso), y añadir una indicación de la equivalencia al gráfico. La identificación de múltiples recursos o representaciones de la misma entidad del mundo real también se conoce como mapeado ontológico, coincidencia ontológica, o alineación ontológica.

Un recurso puede ser un recurso web desreferenciable por un URI, y puede ser una representación (descripción) de una entidad descrita por datos del gráfico. Una entidad particular puede estar representada por uno o más recursos.

Retardar el procesado de reconciliación puede incluir realizar los procesos de identificar la equivalencia semántica entre recursos y añadir al gráfico enlaces que indiquen las equivalencias semánticas, y luego esperar uno de un conjunto predeterminado de eventos de disparo, o una decisión hecha por un programador de tareas en respuesta al estado (tal como el uso/carga) de recursos informáticos en el sistema, antes de reanudar dichos procesos.

En realizaciones de la presente invención, seleccionar el subconjunto inicial puede incluir determinar qué nodo tiene el número más alto de accesos de lectura en operaciones de representación gráfica, e incluir los nodos en el subconjunto inicial como los nodos más visitados.

El tamaño del subconjunto inicial puede ser controlable por un administrador de sistema, o puede ser preestablecido, y puede ser, por ejemplo, un número establecido de nodos o una proporción establecida de nodos en el gráfico. Una operación de representación gráfica puede incluir visitar/leer nodos de gráfico de manera particular; eventualmente todos los nodos podría ser visitados donde el método de visitar puede afectar a la eficiencia del algoritmo de representación. Ventajosamente, seleccionando los nodos más visitados en el subconjunto inicial, el procesado de reconciliación se completa entre el nodo particular y los nodos de consulta más probable. Por lo tanto, es probable que los nodos leídos como parte de una consulta al gráfico hayan tenido procesado de reconciliación realizado con otros nodos en el gráfico. Además, seleccionar el subconjunto inicial también puede incluir determinar qué nodo tiene la probabilidad más alta de tener un acceso de lectura como el nodo siguiente después de uno de los nodos más visitados en una operación de representación gráfica.

- Ventajosamente, en este método particular, la aparición conjunta con los nodos más visitados también se usa como una base para selección en el subconjunto inicial. Esto amplía el rango de nodos que habrán tenido procesado de reconciliación realizado con una alta proporción de los otros nodos cuando sean consultados, y la extensión se dirige de manera que sea simpática al mecanismo con el que se hacen los accesos de lectura a los nodos. Usando una
- 5 regla de asociación de apariciones conjuntas para identificar los nodos que muy probablemente serán visitados después de uno de los nodos más visitados en una operación de representación gráfica, el procesado de reconciliación realizado para los nodos más visitados también puede ser realizado (propagarse) para los nodos identificados, como parte del subconjunto inicial.
- 10 En una implementación alternativa, puede ser que los nodos que tengan la probabilidad más alta de tener un acceso de lectura como el nodo siguiente después de que uno de los nodos más visitados en una operación de representación gráfica sean identificados como un subconjunto adicional para el que se retarda el procesado de reconciliación. Por ejemplo, subconjuntos adicionales pueden estar asociados con un rango de probabilidades de nodos que tienen un acceso de lectura como el nodo siguiente después de uno de los nodos más visitados en una
- 15 operación de representación gráfica, y los nodos pueden ser asignados a otro subconjunto dependiendo de dicha probabilidad.
- En implementaciones en las que se ha determinado que, en un tiempo u otro, entre cada nodo y otro nodo se debe realizar procesado de reconciliación, el subconjunto adicional de los nodos son todos los nodos del gráfico que no son elementos del subconjunto inicial. Sin embargo, puede ser que en el subconjunto adicional de nodos no se realice necesariamente todo el procesado de reconciliación entre ellos y el nodo particular en una sola rutina, sino que se puede dividir en subconjuntos adicionales, con retardos entre el procesado de reconciliación para cada subconjunto adicional.
- 20 Realizar procesado de reconciliación entre cada nodo y cada uno de los otros nodos es deseable porque permite alinear una con otra varias aplicaciones que utilizan o analizan los datos en el gráfico y facilita la comunicación entre ellas. Por ejemplo, un recolector de datos puede seguir los enlaces de equivalencia producidos por procesado de reconciliación para navegar a través de una web de datos potencialmente sinfín.
- 25 Opcionalmente, el procesado de reconciliación del subconjunto adicional de los nodos puede ser retardado y realizado durante intervalos de tiempo cuando la supervisión de recursos informáticos indica que los recursos informáticos estarán en reposo en caso contrario.
- Puede ser que el procesado de reconciliación se realice solamente en el tiempo de reposo del sistema. Opcionalmente, puede ser que el procesado de reconciliación del nodo particular con el subconjunto inicial se realice en un tiempo determinado por eventos en el gráfico, por ejemplo, en el tiempo de introducción de un nuevo elemento de datos (tal como una tripla RDF) en el gráfico, y que el procesado de reconciliación entre el subconjunto adicional y el nodo particular se retarde hasta dicho tiempo de reposo del sistema. El tiempo de reposo del sistema se usa aquí como notación conveniente de un rango de estados operativos en el que hay capacidad de manejar la carga de
- 35 procesado extra impuesta por el procesado de reconciliación al recurso responsable del procesado de reconciliación. Por ejemplo, puede ser que el procesado de reconciliación del subconjunto adicional sea realizado por el procesador cuando no haya otras tareas pendientes en el procesador. Tales decisiones puede tomarlas un programador de carga de trabajo centralizado (que tiene la funcionalidad de un módulo de control de procesado de reconciliación) o localmente en el procesador propiamente dicho. Es decir, puede ser que la decisión de cuándo realizar procesado de reconciliación entre el nodo particular y el subconjunto adicional de nodos se tome dependiendo del estado operativo del recurso de cálculo responsable de realizar dicho procesado.
- 40 Los métodos que realizan la presente invención pueden incluir además: añadir una identificación del nodo particular a una lista; mantener la identificación del nodo particular en la lista hasta que el procesado con el subconjunto adicional de los nodos del gráfico haya sido realizado, después de lo que la identificación del nodo particular se quita de la lista; y cuando se recibe una consulta al gráfico que identifica uno o más nodos a leer, verificar cuál del único o los varios nodos a leer es identificado en la lista, y por cada uno del único o los varios nodos a leer que sea identificado en la lista, realizar procesado de reconciliación con cada nodo del gráfico.
- 45 Tal lista podría mantenerse a nivel central, por una entidad que tenga la función de un controlador de reconciliación (tal como un módulo de control de procesado de reconciliación) o controlador de sistema. Alternativamente, puede ser que la lista se realice dentro de una arquitectura distribuida, y que sea realizada por más de un nodo de un sistema distribuido que trabajen en cooperación uno con otro. En algunas implementaciones, puede ser deseable que el procesado de reconciliación sea completado con respecto a un nodo particular antes de ser devuelto como parte de una respuesta a una consulta, o leído como parte de una respuesta a una consulta. Ventajosamente, mantener una lista de los nodos para los que todavía no se ha realizado un procesado de reconciliación completo (es decir, procesado de reconciliación con otros nodos en el gráfico) permite la fácil identificación del procesado de reconciliación que se tiene que realizar antes de devolver una respuesta a una consulta. La identificación del nodo en la lista puede ser, por ejemplo, un nombre o un URI. La identificación de un nodo particular puede permanecer en
- 50 la lista hasta que el procesado de reconciliación con el subconjunto adicional de nodos haya sido iniciado o completado, dependiendo de la implementación.
- 55
- 60
- 65

Adicional o alternativamente, por cada nodo del gráfico no incluido en el subconjunto inicial con el que se lleva a cabo procesado de reconciliación, un método que realiza la presente invención puede incluir añadir una identificación del nodo particular a una lista de nodos no reconciliados en los metadatos asociados, y mantener dicha identificación en dicha lista hasta que se lleve a cabo el procesado de reconciliación con el nodo particular, después de lo que la identificación del nodo particular se quita de dicha lista; y cuando se recibe una consulta al gráfico que identifica el único o los varios nodos a leer, por cada uno de los nodos a leer, realizar procesado de reconciliación entre dicho nodo y cada uno de los nodos identificados en la lista de nodos no reconciliados en los metadatos asociados.

Hay implementaciones en las que cada uno de los nodos tiene metadatos asociados, que pueden estar almacenados en el nodo o en una posición enlazada a/de el nodo o localizable en base a una identificación del nodo. Cada nodo tiene sus propios metadatos individuales asociados. Los metadatos pueden incluir una lista de nodos que han sido reconciliados con (= se realizó procesado de reconciliación entre ellos propiamente dichos y) el subconjunto inicial de nodos, pero todavía no con el nodo en cuestión. Ventajosamente, esto permite realizar el control del procesado de reconciliación con el subconjunto adicional de nodos localmente en base a información disponible en el nodo o accesible directamente desde el nodo.

Aunque no es esencial en realizaciones de la presente invención, las implementaciones pueden incluir sistemas en los que el gráfico está almacenado en una red distribuida de recursos informáticos. La red distribuida de recursos informáticos (nodos de almacenamiento) puede incluir un sistema de más de una unidad de almacenamiento distinta en comunicación entre sí. Un paradigma de comunicación ejemplar es entre iguales (P2P), por lo tanto puede ser que la red distribuida de recursos informáticos sea una red entre iguales de nodos de almacenamiento. P2P es una arquitectura distribuida que divide tareas o cargas de trabajo entre iguales. Los iguales (nodos o procesos de almacenamiento individuales) son participantes equipotentes, igualmente privilegiados, en la aplicación. Cada igual está configurado para hacer que una porción de sus recursos, tal como potencia de procesado, almacenamiento en disco o anchura de banda de red, esté directamente disponible para otros participantes en la red, sin necesidad de coordinación central por servidores o hosts estables. Puede considerarse que los iguales son tanto proveedores como consumidores de recursos, en contraposición a un modelo cliente-servidor tradicional donde los servidores suministran y los clientes consumen. Ventajosamente, un sistema P2P puede mantener grandes grupos de nodos de almacenamiento intercambiando mensajes con un costo de comunicación logarítmico.

En términos del proceso actual de realizar reconciliación entre dos nodos, en realizaciones de la presente invención, el procesado de reconciliación del nodo particular con otro nodo puede incluir: obtener un valor de semejanza agregando los valores resultantes de realizar una pluralidad de diferentes cálculos de semejanza, u obtener un solo valor resultante de realizar un cálculo de semejanza, donde un cálculo de semejanza es un cálculo que genera un valor que representa la probabilidad de que el nodo particular y el otro nodo sean semánticamente equivalentes; y si el valor de semejanza excede de un umbral de semejanza predeterminado, añadir al gráfico un enlace de equivalencia que denota una relación de equivalencia entre el nodo particular y el otro nodo.

Ventajosamente, las realizaciones de la presente invención reducen el requisito de realizar todo el procesado de reconciliación para un nodo particular de una vez, o en una sola rutina. Con la flexibilidad añadida del acercamiento incremental al procesado de reconciliación empleado en realizaciones de la presente invención, se puede emplear un algoritmo más robusto o un conjunto de algoritmos en el procesado de reconciliación propiamente dicho, sin impactar negativamente en el rendimiento general del sistema. Para ello, las realizaciones de la presente invención pueden ejecutar una pluralidad de algoritmos cada uno de los cuales dé un valor de semejanza, o ejecutar un único algoritmo robusto, que dé un valor de semejanza, y agregar estos valores para obtener un valor de semejanza general para comparación con un umbral. Puede ser que cada algoritmo (o cálculo de semejanza) dé un sí/no (1/0) acerca de si dos nodos son o no semánticamente equivalentes, y el umbral es la proporción de resultados afirmativos requeridos del número total de cálculos de semejanza realizados. Alternativamente, se puede usar una media ponderada de un número de cálculos de semejanza para obtener un valor de semejanza.

Dependiendo de la manera en que se implemente una realización de la presente invención, puede ser que el procesado de reconciliación se realice simultáneamente en más de un recurso de cálculo dentro de la red distribuida de recursos informáticos, entre el nodo particular y los nodos almacenados en dicho recurso de cálculo y pertenecientes al subconjunto de nodos con respecto a los que se esté realizando el procesado de reconciliación con el nodo particular.

Los sistemas de almacenamiento distribuidos, como los que se podría emplear para almacenar un gráfico de datos, imponen una segmentación en los datos por su naturaleza. Ventajosamente, las realizaciones de la presente invención pueden aprovechar dicha segmentación realizando reconciliación en paralelo para un nodo particular realizando procesado de reconciliación en cada recurso de cálculo usado para almacenamiento de nodos entre el nodo particular y los nodos de gráfico almacenados en dicho recurso de cálculo. Puede ser que se realice cierta segmentación ortogonal de los datos, de modo que cada recurso de cálculo no realice procesado de reconciliación entre el nodo particular y cada uno de los nodos almacenados en él en una rutina, sino que divida los nodos, por ejemplo, en base a la frecuencia con que los nodos son visitados en operaciones de representación gráfica. Esta

segmentación ortogonal proporciona mayor flexibilidad en implementaciones de procesado en paralelo. Puede ser que el procesado de reconciliación entre el nodo particular y un subconjunto inicial se realice en el centro, y que luego se realice el procesado de reconciliación entre el nodo particular y el subconjunto adicional, y posiblemente también se programe, localmente.

5 Las realizaciones de la presente invención pueden mejorar o realizar su funcionalidad mediante el almacenamiento de ciertas estadísticas/valores/propiedades/parámetros/mediciones como metadatos almacenados en base por nodo. Las realizaciones de la presente invención pueden incluir almacenar, para cada nodo, como metadatos asociados, un valor que represente los números de accesos de lectura realizados al nodo en operaciones de representación gráfica, y un valor que represente el número de veces que se leyó cada nodo con el que el nodo esté enlazado en el gráfico como el nodo siguiente en una operación de representación gráfica.

15 En realizaciones de la presente invención, el conocimiento, los hechos y/o las declaraciones se representan con un gráfico de nodos y bordes, donde los nodos son las entidades descritas o representadas, y los bordes son las relaciones entre dichas entidades. Las realizaciones de la presente invención pueden estar configuradas para almacenar datos gráficos directamente, es decir, como nodos y bordes. Sin embargo, puede ser que se emplee alguna otra estructura de datos subyacente.

20 Como una estructura de almacenamiento de datos subyacente ejemplar, puede ser que los datos en el gráfico estén codificados como triplas incluyendo cada una un sujeto, un predicado, y un objeto, y los nodos del gráfico son los sujetos y los objetos de las triplas, y el predicado de una tripla denota un enlace entre el sujeto y el objeto de la tripla.

25 Opcionalmente, las triplas pueden ser triplas del Marco de Descripción de Recursos (RDF). En todo este documento, se deberá entender que donde se hacen referencias específicas a “tripla(s) RDF”, es una forma de tripla ejemplar, conforme al estándar RDF. Además, las referencias a “tripla(s)” incluyen la posibilidad de que la tripla en cuestión sea una tripla RDF. Igualmente, los procesadores RDF explicados en otro lugar en este documento son ejemplos de procesadores usados para interacción entre el wrapper API y los elementos de datos almacenados.

30 El Marco de Descripción de Recursos es un método general para la descripción conceptual o el modelado de información que es un estándar para redes semánticas. Estandarizar el modelado de información en una red semántica permite la interoperabilidad entre aplicaciones que operan en una red semántica común. RDF mantiene un vocabulario sin semántica formal ambigua, proporcionando el esquema RDF (RDFS) como un lenguaje para describir vocabularios en RDF.

35 Opcionalmente, cada uno del único o los varios elementos de la tripla (siendo elemento el predicado, el objeto, o el sujeto) es un identificador de recursos uniformes (URI). RDF y otros formatos de tripla son premisa de la noción de identificar cosas (es decir, objetos, recursos o casos) usando identificadores web como URIs y describir las ‘cosas’ identificadas en términos de simples propiedades y valores de propiedad. En términos de la tripla, el sujeto puede ser un URI que identifique un recurso web que describa una entidad, el predicado puede ser un URI que identifique un tipo de propiedad (por ejemplo, color), y el objeto puede ser un URI que especifique la instancia particular de dicho tipo de propiedad atribuido a la entidad en cuestión, en su encarnación de recursos web. El uso de URIs permite que las triplas representen declaraciones simples, con relación a recursos, como un gráfico de nodos y arcos que representan los recursos, así como sus respectivas propiedades y valores. Un gráfico RDF puede ser consultado usando el protocolo SPARQL y el lenguaje de consulta RDF (SPARQL). Fue estandarizado por el Grupo de Trabajo de Acceso a Datos RDF (DAWG) del World Wide Web Consortium, y se considera una tecnología web de clave semántica. SPARQL permite que una consulta conste de configuraciones de triplas, uniones, desuniones, y configuraciones opcionales.

50 Las triplas proporcionan codificación de datos gráficos caracterizando los datos gráficos como una pluralidad de expresiones de sujeto-predicado-objeto. En ese contexto, el sujeto y el objeto son nodos de gráfico de los datos gráficos, y como tales son entidades, objetos, casos, o conceptos, y el predicado es una representación de una relación entre el sujeto y el objeto. El predicado afirma algo acerca del sujeto proporcionando un tipo de enlace especificado al objeto. Por ejemplo, el sujeto puede denotar un recurso web (por ejemplo, mediante un URI), el predicado denota un rasgo particular, característica o aspecto del recurso, y el objeto denota una instancia de dicho rasgo, característica o aspecto. En otros términos, una recogida de declaraciones de triplas representa intrínsecamente datos gráficos direccionales. El estándar RDF proporciona estructura formalizada para tales triplas.

60 El procesado de reconciliación entre un nodo y un subconjunto inicial de nodos puede ser iniciado a la entrada de una nueva tripla al gráfico. Además, puede ser condicional sobre si el sujeto de la nueva tripla es en sí mismo nuevo (en cuyo caso el procesado de reconciliación se inicia para dicho ‘nodo particular’), o no (en cuyo caso se decide que no se requiere procesado de reconciliación).

65 Realizaciones de un aspecto particular de la presente invención proporcionan un controlador de base de datos para realizar reconciliación de recursos representados en un gráfico de datos como nodos; incluyendo el controlador de base de datos: un módulo de selección configurado para seleccionar un subconjunto inicial de los nodos del gráfico

5 para los que se ha de realizar procesado de reconciliación con un nodo particular; un módulo de procesado de reconciliación configurado para realizar procesado de reconciliación entre el nodo particular y cada uno del subconjunto inicial de nodos; un módulo de control de procesado de reconciliación configurado para retardar el procesado de reconciliación de otro subconjunto de los nodos del gráfico para el que se ha de realizar reconciliación con el nodo particular, donde el procesado de reconciliación del subconjunto adicional de los nodos es retardado y realizado durante intervalos de tiempo cuando la supervisión de recursos informáticos responsables del procesado de reconciliación indica que dichos recursos informáticos estarían en reposo en caso contrario.

10 El controlador de base de datos podría realizarse como un controlador centralizado en un solo recurso de cálculo, como un controlador centralizado por un número de recursos informáticos que cooperan, o como un controlador entre una pluralidad de controladores equivalentes cada uno en un recurso de cálculo respectivo en un sistema de almacenamiento distribuido. Por ejemplo, puede ser que el controlador de base de datos lo facilite un programa que se ejecute en un recurso de cálculo en el sistema de almacenamiento, y que uno o más recursos informáticos distintos también ejecuten programas equivalentes de modo que la base de datos sea accesible mediante una pluralidad de controladores de base de datos.

15 Realizaciones de otro aspecto de la presente invención proporcionan un sistema de almacenamiento de datos para almacenar un gráfico de datos en el que los recursos se representan como nodos del gráfico, incluyendo el sistema de almacenamiento de datos: una pluralidad de unidades de almacenamiento configurada cada una para almacenar un segmento de datos del gráfico de datos; y un controlador de base de datos como se ha descrito anteriormente y/o en otro lugar como una realización de la invención. Naturalmente, las unidades de almacenamiento de datos son ejemplos de recursos informáticos, y pueden tener funcionalidad de procesado y funcionalidad de control/gestión además de almacenamiento.

20 Las unidades de almacenamiento pueden ser recursos informáticos, por ejemplo, cada una puede incluir una unidad de almacenamiento, además de un procesador, memoria, y/o componentes adicionales tal como una tarjeta de interfaz de red, una placa madre, dispositivos de entrada/salida.

25 Realizaciones de otro aspecto de la presente invención proporcionan un programa de ordenador que, cuando es ejecutado por un ordenador, hace que el ordenador realice un método que realiza la presente invención. Además, realizaciones de otro aspecto de la presente invención incluyen un programa de ordenador, que, cuando es ejecutado por uno o más ordenadores, hace que el único o los varios ordenadores funcionen como un controlador de base de datos que realiza la presente invención. Los programas de ordenador que realizan la presente invención pueden estar almacenados en un medio de almacenamiento legible por ordenador, tal como un medio de almacenamiento no transitorio, y se pueden proporcionar como un solo programa de ordenador o como una serie de programas secundarios.

30 Realizaciones de la presente invención implementan un acercamiento para realizar procesado de reconciliación que tiene uno o varios de los efectos ventajosos siguientes: reducir el esfuerzo computacional de la reconciliación; descomponer el procesado en trozos más pequeños para permitir una mejor programación de las tareas de reconciliación; aprovechar el almacenamiento de datos distribuido para permitir la segmentación ortogonal de los datos para aumentar el rendimiento en paralelo y mejorar la escalabilidad; permitir la ejecución de varias medidas de semejanza al mismo tiempo y en recursos de ordenador dispersados; es guiado por un modelo de probabilidad para realizar reconciliación.

35 Ahora se describirán características preferidas de la presente invención, puramente a modo de ejemplo, con referencia a los dibujos acompañantes, en los que:

40 La figura 1 es una ilustración esquemática del software desplegado en un servidor en un sistema de almacenamiento de datos que realiza la presente invención.

45 La figura 2 ilustra una arquitectura de sistema de una realización de la presente invención en la que parte de la funcionalidad de control está centralizada.

50 La figura 3 es un diagrama de flujo que ilustra el procesado realizado al añadir una nueva tripla a un almacenamiento de datos subyacente en una realización de la presente invención.

55 La figura 4 es un diagrama de flujo que ilustra el procesado realizado al actualizar el gráfico extendido y realizar procesado de reconciliación en una realización de la presente invención.

60 La figura 5 es un diagrama de flujo que ilustra un proceso ejemplar para realizar procesado de reconciliación en una realización de la presente invención.

65 La figura 6 es un diagrama de flujo que ilustra un proceso ejemplar para recibir y manejar una consulta en una realización de la presente invención.

La figura 7 es un diagrama de flujo que ilustra un proceso ejemplar para llevar a cabo reconciliación exhaustiva para un nodo en una realización de la presente invención.

5 La figura 8 es un diagrama de flujo que ilustra un proceso ejemplar para llevar a cabo procesado de reconciliación "en nodo" en una realización de la presente invención.

La figura 9 es una ilustración conceptual de paralelización de procesado de reconciliación.

10 La figura 10 es una ilustración conceptual de la segmentación de datos habilitada por distribución física de datos y utilizando valores almacenados como metadatos.

15 La figura 1 ilustra la arquitectura del software desplegado en cada servidor 10. En este ejemplo particular, el sistema de almacenamiento está compuesto por un conjunto de servidores que comunican a través de un sistema de paso de mensajes. En el ejemplo ilustrado en la figura 1, los servidores implementan un modelo de capas múltiples donde cada capa explota y usa capacidades de las capas situadas debajo. La arquitectura consta, partiendo de la capa superior, de una capa semántica 11, una capa RDF 12, una capa de adaptador de tripla 14, una capa KVS 15, y una capa de comunicación 16 con un detector de fallo 13 que se extiende junto a la capa de adaptador de tripla 14 y la capa KVS 15. Naturalmente, como será claro por la descripción que sigue, la realización aquí explicada usa triplas RDF como la estructura de datos subyacente para almacenar el gráfico de datos. Ésta no es de ningún modo la única estructura de datos subyacente que se puede usar en realizaciones de la invención, pero se ha seleccionado aquí al objeto de presentar una implementación ejemplar de realizaciones de la invención. Además, se emplea un formato KVS (almacenamiento de valor clave), en el que las triplas RDF propiamente dichas se usan como claves para acceder a más datos (incluyendo metadatos) en un almacenamiento de valor asociado. De nuevo, el formato KVS se selecciona para presentar una implementación ejemplar de realizaciones de la invención, y será evidente a los expertos que las realizaciones de la invención no se limitan a este formato concreto.

Los párrafos siguientes describen el papel de cada capa y su interacción.

30 1) *Capa semántica*: la capa semántica 11 es la interfaz a aplicaciones de cliente que interactúan con usuarios finales y está configurada para enviar consultas de usuario al procesador RDF. En esta capa, los datos no se consideran como triplas RDF individuales, sino a un nivel conceptual o incluso a nivel modelo (como grupos de triplas RDF relacionados). Las operaciones de procesado semántico tales como procesado de reconciliación de datos y representación de propiedad se realizan en esta capa. El análisis ontológico de mayor refinamiento técnico, tal como inferencia basada en normas e inferencia basada en lógica descriptiva, también se puede habilitar cuando se necesite un análisis semántico profundo. En la implementación ejemplar, se puede realizar inferencias de nivel alto a través de Jena y OWL2 RL/RDF. Sin embargo, la capa semántica 22 no se limita a implementaciones específicas y permite conectar fácilmente otros motores de inferencia. La capa semántica 11 proporciona la funcionalidad de control del módulo de selección, el módulo de procesado de reconciliación, y el módulo de control de procesado de reconciliación mencionado en otro lugar en este documento.

40 2) *Capa RDF*: la capa RDF 12 puentea la capa de datos subyacente y la capa semántica. Esto se puede implementar con una API tal como Jena Graph API. La capa RDF 12 está configurada para usar métodos que operan directamente con triplas, por ejemplo añadir/quitar triplas y recuperar triplas usando consultas RDF SPARQL. También en la capa RDF 12 hay un motor SPARQL, responsable de convertir la consulta basada en texto a un conjunto de operaciones. Estas operaciones incluyen operaciones de búsqueda a nivel bajo, mecanismos de filtración y unión, entre otros. Algunas de estas funcionalidades se implementan dentro del motor SPARQL en esta capa, mientras que la búsqueda de datos a nivel bajo la facilita la capa de adaptador de tripla subyacente 14.

50 3) *Capa de adaptador de tripla*: la capa de adaptador de tripla 14 es responsable de convertir triplas a una clave que puede ser almacenada en la capa KVS 15. Esta clave deberá ser única y también puede ser comparable con otras claves con el fin de implementar un orden (en el que cada clave es mayor, menor o igual a cualquier otra clave generada a partir de una tripla) sobre claves. La capa de adaptador de tripla 14 proporciona a la capa superior tres operaciones simples: añadir una nueva tripla, sacar una tripla existente, y hallar un conjunto de triplas según criterios específicos. La operación *hallar* es específica de RDF, recibe tres parámetros, que pueden ser un valor específico o *nulo*. Por ejemplo, *hallar (sujeto; nulo; nulo)* devuelve todas las triplas que tienen un sujeto específico (definido). Esta capa usa la interfaz KVS subyacente para almacenar, quitar y consultar datos.

60 4) *Capa KVS*: la capa KVS 15 implementa un almacenamiento distribuido de valores clave ordenados. Ofrece una interfaz de mapa hash ordenado regular para añadir un par de valores clave, para obtener o quitar un valor clave (dada alguna clave) y, adicionalmente, para ejecutar consultas de rango sobre el KVS ordenado. El método de consulta de rango recibe una clave mínima prefijo y una clave máxima prefijo, y devuelve un iterador sobre todos los pares de valores clave contenidos dentro del rango dado. Esta capa es responsable de la distribución y recuperación de los datos de otros servidores. Esta capa usa la capa de comunicación subyacente 16 para comunicar con los otros servidores pasando mensajes.

65 5) *Detector de fallo*: el detector de fallo 13 es responsable de la ejecución de vigilantes que detectarán un fallo de

servidor y notificarán a la capa de almacenamiento los servidores supervivientes (sin fallo). También usa la capa de comunicación 16 para intercambiar mensajes entre otros servidores y producir un conjunto de procesos vivos, también llamado una vista del sistema. Cuando esta vista cambia, se le notifica a la capa KVS 15 de modo que, por ejemplo, pueda disparar la recuperación de fallos. El detector de fallo 13 es muy opcional en el contexto de las realizaciones de la presente invención, y proporciona funcionalidad que amplía la funcionalidad de las realizaciones de la invención.

6) *Capa de comunicación*: la capa de comunicación 16 es un mecanismo por el que se pueden pasar mensajes entre servidores (ejemplos de unidades de almacenamiento o dispositivos informáticos mencionados en otro lugar en este documento). Por ejemplo, la capa de comunicación 16 puede estar configurada para exportar primitivos para enviar y recibir mensajes a/de otros servidores. El lector experto apreciará que se puede usar muchas implementaciones alternativas para esta finalidad.

La figura 2 ilustra una arquitectura alternativa, en la que una aplicación de cliente 30 comunica con un solo punto de contacto centralizado mediante un procesador RDF centralizado 22. El procesador RDF 22 tiene la funcionalidad de la capa RDF 12, pero en el ejemplo de la figura 2 se implementa a un nivel centralizado, más bien que local. El procesador de reconciliación 21 (que combina la funcionalidad del módulo de selección, el módulo de procesado de reconciliación, y el módulo de control de procesado de reconciliación mencionados en otro lugar en este documento) también se implementa a un nivel centralizado en el ejemplo de la figura 2. El procesador de reconciliación 21 está configurado para controlar y realizar procesado de reconciliación para un gráfico almacenado en la pluralidad de unidades de almacenamiento de datos 23. La pluralidad de unidades de almacenamiento de datos 23 están configuradas al menos para almacenar un segmento de un gráfico de datos. Además, las unidades de almacenamiento de datos 23 pueden estar configuradas para realizar las funciones de una o varias de la capa de adaptador de tripla 14, la capa KVS 15, la capa de comunicación 16, y el detector de fallo 13 descritos anteriormente. Las figuras 1 y 2 ilustran que varias funciones de realizaciones de la presente invención se pueden llevar a cabo a un nivel local o a un nivel centralizado, dependiendo de los requisitos del sistema en el que se implemente la realización.

KVS se emplea en la realización ejemplar de la figura 1 (y es igualmente aplicable a la figura 2) como acercamiento para escalar el almacenamiento de datos RDF, es decir, el almacenamiento del gráfico de datos. Sin embargo, las triplas RDF propiamente dichas no están preparadas para KVS. Un primer paso es definir las tuplas de valores clave que un KVS estándar puede utilizar convenientemente. A continuación se expone una estructura de datos formalizada ejemplar para un almacenamiento distribuido (gráfico) de datos.

Definición 1: Tripla RDF. Una tripla RDF t , (s,p,o) (denotado como SPO) define la relación entre un sujeto (s) y el objeto (o). La relación se refiere a usar un predicado (p). El sujeto (s) y el objeto (o) son nodos en el gráfico, y el predicado es un arco.

Tripla es la estructura de datos fundamental en RDF. Como tal, las triplas son una estructura de datos subyacente ejemplar para almacenar una base de datos de gráfico (que se puede considerar que es una forma almacenada del gráfico) en realizaciones de la presente invención. Sin embargo, la presente invención también puede realizarse o aplicarse a realizaciones en las que se emplee una estructura de datos subyacente distinta de triplas RDF. El objeto de una tripla puede ser, a su vez, el sujeto de otras triplas o de tipos de datos concretos tal como cadena, entero, etc. Un gráfico es una presentación natural de triplas RDF. Un gráfico RDF es un gráfico etiquetado dirigido. Los vértices (nodos) del gráfico son sujetos u objetos de triplas y los bordes son predicados. En un gráfico RDF, ambos nodos y bordes también se denominan recursos RDF.

Definición 2: Gráfico RDF. Sea T un conjunto de triplas RDF. La representación gráfica de T es $G = (V,E)$ donde $V = \{v | \exists t \in T. (v \in \text{sujeto}(t) \cup \text{objeto}(t))\}$ y $E = \{(v_s, v_o) | (s,p,o) \in T\}$. V se define como un conjunto de nodos (o vértices) como sujetos u objetos de triplas RDF mientras que E es un conjunto de bordes (o arcos) entre nodos generalmente especificados como un par de nodos $(v_1, v_2), v_1, v_2 \in V$.

En términos generales, KVS es un modelo de datos no relacional que permite almacenar datos sin esquema. Los beneficios clave de KVS son la simplicidad y la escalabilidad. Los elementos de datos en KVS son tuplas de valores clave (k,v) , donde k es un identificador único y v la carga que k identifica.

Definición 3: Tupla K/V (valor clave) triplicada. Dado una tripla t , la tupla K/V triplicada es

$$\{\langle k_i, metadata \rangle \mid k_i \in \{\langle s, p, o \rangle\} \cup \{\langle p, o, s \rangle\} \cup \{\langle o, s, p \rangle\}\}$$

donde los metadatos guardan metadatos de tripla como el valor de una tupla de valores clave.

Como un ejemplo de un almacenamiento de triplas como un formato de datos subyacente para realizaciones de la presente invención, cada tripla de la forma SPO se replica generando dos variantes semánticamente idénticas, pero

sintácticamente diferentes en forma de (p,o,s) (denotado como POS) y (o,s,p) (denotado como OSP). Esto bajo la consideración de rendimiento de consulta y recuperación de fallo, y con vistas a proporcionar un mecanismo para almacenar varias formas de metadatos utilizados en algunas realizaciones de la invención.

5 La consulta de datos RDF se efectúa normalmente mediante consultas de rango, sustituyendo elementos en triplas por comodines. Por ejemplo `SELECT(?,?,?)` recupera todas las triplas en el almacenamiento de datos mientras que `SELECT(?,?,o)` devuelve todas las triplas con o como objeto. La réplica de datos permite clasificar datos según no solamente los sujetos sino también los predicados y los objetos. Esto mejora la ejecución de la consulta.

10 Las triplas son almacenadas en un KVS distribuido, en esta implementación específica, con cada tripla y sus variantes como una clave y los metadatos relacionados como un valor. Tal acercamiento facilita el modelado y la gestión de metadatos flexibles. Como se ha mencionado previamente en este documento, se podría haber usado cualquier otro almacenamiento basado en gráfico, preferiblemente las realizaciones de la presente invención se implementan de tal forma que los metadatos puedan ser almacenados y accedidos mediante el almacenamiento basado en gráfico. En las realizaciones de la invención, los metadatos pueden ser usados para registrar frecuencias de visita a nodos y apariciones conjuntas de pares de nodos como se explica a continuación. A su vez, los metadatos se usan para seleccionar qué nodos de gráfico incluir en el subconjunto inicial de nodos, y cuáles pueden tener procesado de reconciliación con un nodo particular retardado.

20 El conjunto tanto de triplas originales de la forma SPO como sus réplicas de la forma POS y OSP se clasifica y divide en n segmentos, donde n es el número de servidores (o unidades de almacenamiento) que se usarán para almacenar los datos, con cada servidor responsable para un segmento de datos. Se puede usar una función hash para mapear segmentos de los datos a servidores.

25 Con el fin de localizar eficientemente el segmento donde se deberá colocar una tripla específica o una de sus variantes, se puede usar un Trie (un árbol clasificado con complejidad de búsqueda vinculada al tamaño de cadena a indexar), donde cada hoja del Trie corresponde a un segmento de datos. Por ejemplo, se puede implementar un mapeado de dos etapas desde un elemento de datos al servidor. Todas las triplas existentes son organizadas y clasificadas usando el Trie, disponiéndose todo el conjunto de triplas en orden alfabético. El hallazgo del segmento de datos correcto se efectúa entonces siguiendo el prefijo asociado con cada nodo en el Trie. Después de recuperar las IDs de segmento, se puede emplear hashing consistente como una herramienta para mapear segmentos a máquinas servidoras. Poniendo en cascada los dos mapeados conjuntamente, una función (γ) que es capaz de hallar a cada tripla su máquina de almacenamiento destino: $\gamma: t \rightarrow \text{IDmachine}$ con implementación $\gamma = \text{hashing consistente (búsqueda de trie (t))}$. Esta función mapea un elemento de datos a un segmento ID usando la estructura Trie y a continuación el segmento ID es mapeado a una ID de máquina usando hashing consistente. Este método se usa para almacenar todas las versiones de triplas, como se describe en los párrafos siguientes.

La adición de una tripla incluye los pasos de localizar el segmento de datos en el espacio de datos ordenado e insertar la tupla de valores clave en la máquina correspondiente (usando mapeado γ ()). Un algoritmo ejemplar para introducción de tripla se detalla a continuación en algoritmo 1.

Algoritmo 1: addTriple(t)

Requerir: $\gamma()$

45 $m_1 \leftarrow \gamma(t.SPO), m_2 \leftarrow \gamma(t.POS), m_3 \leftarrow \gamma(t.OSP)$
 almacenar ((t.SPO,metadatos), m_1,m_2,m_3) en m_1
 almacenar ((t.PPO,metadatos), m_1,m_2,m_3) en m_2
 almacenar ((t.OSP,metadatos), m_1,m_2,m_3) en m_3

50 Los argumentos pasados a una máquina de almacenamiento incluyen la tripla o una variante de la tripla y los identificadores (m_1, m_2, m_3) de todas las máquinas implicadas (para variantes) que se mantienen en una tabla índice de almacenamiento local de servidor para recuperación de fallo. Estos procedimientos aseguran que cada tripla SPO, POS y OSP se almacene en el sistema de tal forma que permita consultas de rango usando como prefijo cualquier combinación de un subconjunto de los datos de tripla.

El mecanismo de localizar una tripla es el mismo para recuperación y borrado. Dado que los datos siempre están ordenados, es posible recuperar IDs de segmento necesarias para manejar consultas de rango RDF. La definición 5 expone cómo se podría realizar una consulta de rango en esta implementación específica.

60 **Definición 5: Consulta de rango.** Dadas dos triplas t_1 y t_2 y segmentaciones de datos D_0, \dots, D_n , una consulta de rango recupera todas las triplas entre t_1 y t_2 inclusive como:

$$\{t \mid t \in D_i \wedge t_1 \in D_i \wedge key(t) \geq key(t_1)\}$$

$$\cup \{t' \mid t' \in D_j \wedge t_2 \in D_j \wedge key(t') \leq key(t_2)\}$$

donde key() convierte una tripla a una clave

$$\cup \{t'' \mid t'' \in D_k \wedge (i < k < j)\}$$

5 en KVS.

Los segmentos que tienen que ser visitados son: el segmento conteniendo t_1 , el segmento conteniendo t_2 , y todos los segmentos entremedio.

10 Una operación de borrar se procesa de manera similar sustituyendo la instrucción “almacenar” en el algoritmo 1 por “borrar” en cada uno de los servidores. La instrucción “borrar” quita la fila que corresponde a la tripla y actualiza la tabla índice respectiva, si es necesario.

15 El acercamiento incremental a la reconciliación de las realizaciones de la presente invención se puede implementar de manera “activada por evento”. Se produce un evento cuando “algo” cambia el estado de un almacenamiento de datos (por ejemplo se introduce una nueva tripla o se recupera una tripla existente), siendo un almacenamiento de datos la versión almacenada del gráfico. A efectos de reconciliación de datos, dos eventos elementales se consideran eventos relevantes que pueden disparar el procesado de reconciliación: addEvent y readEvent.

20 AddEvent es emitido por una operación de introducción de tripla. ReadEvent es emitido por consulta de gráfico. La reconciliación se dispara como una reacción a estos dos eventos. Otros tipos de eventos que pueden impactar en la reconciliación de datos se pueden representar como una serie de addEvents y readEvents y por ello no se definen por separado. Naturalmente, addEvent y readEvent están altamente conectados. Cuando se añade una tripla a un gráfico no vacío, las triplas relacionadas serán visitadas y evaluadas para detectar cualesquiera datos que deban ser

25 reconciliados, haciendo que se emita una serie de readEvent al almacenamiento de datos. Tal relación estrecha se refleja de la manera ejemplar para seleccionar el subconjunto inicial de nodos con los que realizar el procesado de reconciliación con un nodo particular expuesto a continuación. ReadEvent se considera como una operación de representación gráfica. La historia de representaciones se emplea para establecer un subconjunto inicial incluyendo los nodos más visitados y los nodos que tienen una fuerte relación de apariciones conjuntas con los nodos más

30 visitados. El algoritmo para seleccionar el subconjunto inicial es un mecanismo ejemplar para seleccionar el subconjunto inicial, y se basa en el supuesto de que si dos nodos están fuertemente correlacionados (es decir, la probabilidad de visitar el nodo B después de visitar el nodo A en una operación de representación gráfica es muy alta), es altamente probable que el procesado de reconciliación realizado en A también deba realizarse en B en la misma rutina, en vez de retardarse. Siguiendo los recorridos de tales nodos correlacionados usando relaciones de

35 apariciones conjuntas se puede seleccionar un subconjunto inicial de nodos cuya reconciliación sea más inminente que la de otros nodos.

La deseabilidad de tal mecanismo de selección es evidente. Aunque el gráfico está almacenado a través de múltiples servidores, el tamaño de datos en una máquina individual puede ser potencialmente muy grande,

40 denegando una acción de reconciliación exhaustiva en tiempo real. La probabilidad de aparición conjunta entre dos nodos arbitrarios puede ser usada para dividir el gráfico en un gráfico secundario de nodos frecuentemente visitados y los que tienen fuerte correlación con los nodos más frecuentemente visitados (subconjunto inicial), y uno conteniendo menos nodos visitados (subconjunto adicional). Con respecto a éste último, las comprobaciones de reconciliación se posponen hasta que se realice una operación de lectura directa. Por lo tanto, se habilita un

45 acercamiento incremental para reconciliación con respecto a un gráfico de datos grande.

Dividiendo el gráfico en un subconjunto inicial y subconjunto adicional, también es posible superponer la segmentación del gráfico de datos sobre la segmentación de datos físicos ya en posición. Esta segmentación basada en el uso mejora más el procesado en paralelo de la reconciliación de datos.

50 Realizaciones de la presente invención pueden emplear o crear un gráfico extendido, extendido por algunos campos/valores de metadatos particulares que serán elaborados más adelante. Los campos de metadatos particulares proporcionan información que puede ser usada al diferenciar el subconjunto inicial de nodos del subconjunto adicional de nodos.

55 *Definición 4: Gráfico RDF extendido.*

El gráfico RDF extendido es $G_e = (V', E')$ que extiende el gráfico RDF $G = (V, E)$ con metadatos de nodo y borde:

$$V' = \{ \langle v, m_v \mid v \in V \rangle \}, \quad E' = \{ \langle e, m_e \rangle \mid e \in E \},$$

donde V y E son los definidos en la *Definición 2: Gráfico RDF*.

Las triplas son tratadas como la estructura de datos subyacente en esta implementación concreta, y el gráfico RDF extendido definido anteriormente requiere metadatos de nodo local y borde local. Dado que esta implementación guarda tres variantes de cada tripla, los metadatos pueden ser almacenados de la siguiente manera:

* Almacenar metadatos de nodo y borde: m_v se almacena localmente en cada tripla SPO y OSP para nodos que representan sujeto de tripla y objeto de tripla respectivamente. M_e se almacena localmente en cada PSO. Actualizar los metadatos de una tripla existente incluye leer las triplas de metadatos, recalcular el valor, y poner de nuevo el valor.

* Consultar metadatos de nodo y borde: al consultar metadatos acerca de un nodo particular o borde, todas las triplas relevantes son recuperadas y los metadatos se agregan cuando sea apropiado.

Por ejemplo, una indicación de cuántas veces se lee una tripla particular (t) se puede mantener usando una simple tupla de valores clave: (kt, number_of_visits). Si se precisa un módulo funcional para averiguar el número total de visitas al nodo s, deberán recuperarse todas las tuplas de valores clave que tengan (s,?,?) como clave (donde ? corresponde a cualquier cadena). El number_of_visits correspondiente se sumará entonces para obtener la respuesta final.

El gráfico RDF normal se extiende con metadatos que registran las frecuencias de visita a nodos y las apariciones conjuntas de pares de nodos. El uso de frecuencia y aparición conjunta tiene la finalidad de reducir el número de nodos que un proceso de reconciliación tiene que visitar inicialmente y por ello reducir el costo computacional al tiempo de introducción de tripla. Esto permite una técnica de reconciliación incremental que pospone operaciones no urgentes al punto en que sea necesario. Los metadatos de nodo y borde se usan como un mecanismo para indicar qué significativos son los nodos entre otros y cuán estrechamente están conectados los nodos uno a otro.

Se añaden incrementalmente valores numéricos al gráfico normal como resultado de readEvent (consulta disparada por datos). A continuación, el gráfico normal de datos (gráfico RDF) se indica como G y el gráfico extendido G_i , indicando i cuántas iteraciones (disparadas por readEvent y addEvent) han sido aplicadas al gráfico original efectuando la transformación de gráfico incremental.

Definición 6: Frecuencia, aparición conjunta, petición pendiente.

Sea t una tripla (s,p,o) y sean s y o dos nodos resultantes de la representación gráfica de t. La frecuencia, $f(s)$, es el número total de visitas al nodo sujeto s. La aparición conjunta

$$co(s, o) = \frac{m}{f(s)},$$

donde m es el número total de veces que una tripla particular (s,p,o) ha sido visitada/consultada con s como el sujeto.

$L_t = [t_0; \dots, t_i, \dots, t_n]$ es una lista de triplas conteniendo todas las todavía por reconciliar en el nodo. L_t puede estar vacío.

La aparición conjunta entre el nodo s y el nodo o es aproximada como, de todas las visitas en el nodo s, cuántas han conducido a la visita del nodo o. En el nivel de almacenamiento subyacente, los metadatos están asociados con triplas que forman tuplas de valores clave, donde la frecuencia y la petición pendiente se implementan como metadatos de nodo (es decir, (v, f: L_t)) mientras que la aparición conjunta se realiza a través de metadatos de borde como (e, co). Naturalmente, éstas son simplemente técnicas ejemplares para almacenar metadatos requeridos para identificar los nodos más visitados y los nodos visitados después de los nodos más visitados en operaciones de representación gráfica. El lector experto apreciará que se podría usar varios valores diferentes almacenados como metadatos en posiciones diferentes para la misma finalidad.

AddEvent y readEvent son manejados por el sistema de forma diferente. Dado que añadir una tripla a un gráfico da lugar normalmente a que se emita una serie de consultas y potencialmente se generen y añadan nuevas triplas, una instancia de addEvent puede ser considerada como una secuencia de actualizaciones de los valores de frecuencia y aparición conjunta. Teóricamente, tal efecto en cadena terminará solamente si no hay que procesar más nodos.

Sea t una tripla (s,p,o) equivalente a un borde dirigido que va desde el nodo sujeto al nodo objeto. Cada operación

de lectura en el gráfico cambia incrementalmente el valor de frecuencia $f(s)$ del nodo que ha sido visitado y la relación de apariciones conjuntas $co(s,o)$ entre nodos contiguo en G' . La base es que usando la regla de asociación de apariciones conjuntas, se puede decidir si el procesado de reconciliación realizado con respecto a nodo s será/deberá ser propagado a los nodos contiguos que conectan con s , o si el procesado de reconciliación de los nodos contiguos deberá ser retardado.

La figura 3 ilustra un flujo de control en el proceso de añadir una nueva tripla a un gráfico.

Al añadir una tripla a un gráfico vacío, no se necesita reconciliación, S101-S103, 'SÍ' en S102. El gráfico extendido es inicializado entonces como el gráfico RDF sencillo en el paso S104. Al añadir una tripla a un gráfico existente, 'NO' en S102, el gráfico extendido es actualizado en S105 en la medida en que una frecuencia correspondiente del nodo sujeto y la aparición conjunta del borde que conecta el sujeto y el objeto de tripla se incrementan en 1. El procesado de reconciliación se realiza en S105, si es necesario. En esta etapa, puede ser que se seleccione un subconjunto inicial de nodos y que el procesado de reconciliación se realice entre el subconjunto inicial de nodos y el nodo sujeto de la nueva tripla. Un flujo ejemplar de control para el procesado dentro de S105 se expone en la figura 4, como se explica a continuación, y se formaliza en el algoritmo 2. En muchos casos, el procesado de reconciliación puede dar lugar a producir nuevas triplas, 'SÍ' en S106, que a su vez deberá incurrir en actualización de metadatos y nuevo procesado de reconciliación. Esto continuará hasta que no se genere un nuevo conocimiento (tripla). Este proceso se formaliza en el algoritmo 2. Obsérvese que, debido a la naturaleza de KVS, la adición a pares de valores clave existentes simplemente actualiza los valores de las tuplas.

En términos de aparatos que realizan ciertos procedimientos, la selección de un subconjunto inicial puede ser realizada por un módulo de selección. El procesado de reconciliación, incluyendo los cálculos de semejanza, su agregación, y la creación de nuevos enlaces de equivalencia donde sea apropiado, puede ser realizado por un módulo de procesado de reconciliación. La identificación y la programación de procesado de reconciliación adicional (retardado), incluyendo responder a ciertos eventos de disparo e identificar el procesado de reconciliación que deberá ser realizado en respuesta a un evento de disparo, es realizado por un módulo de control de procesado de reconciliación. Cada uno de los módulos puede ser realizado como parte de un controlador de base de datos que puede ser local a una unidad de almacenamiento/servidor/recurso de cálculo en un sistema distribuido, o puede estar centralizado y estar dispuesto en un elemento de hardware centralizado o por hardware/recursos de cálculo/servidores distribuidos que cooperan para crear un controlador centralizado efectivo. Los módulos pueden ser realizados como hardware, o pueden ser módulos de software que operen en un recurso de cálculo/servidor/unidad de almacenamiento, por ejemplo, como parte de un sistema de gestión de base de datos.

La reconciliación de datos/semántica tiene lugar principalmente cuando se afirman nuevos datos de instancia. Al añadir una tripla $t = \langle s,p,o \rangle$, una instancia `addEvent` dispara la realización de reconciliación del sujeto s de la tripla insertada t frente a todo el gráfico. En la práctica, el tamaño de un gráfico frente al que hay que efectuar procesado, puede llegar fácilmente a millones de nodos y ser demasiado grande para permitir una representación exhaustiva. Centrando la selección en un subconjunto inicial con el que se ha de realizar inicialmente reconciliación del nodo particular, en este caso el sujeto de la nueva tripla, se reduce la carga de procesado adelantado y se obtiene una aproximación de reconciliación inicial menos cara.

Algoritmo 2 `doAddEvent` ($t = \langle s, p, o \rangle$)

Require: $G = \langle V, E \rangle$ as the RDF graph

Require: `addTriple()` (Algorithm 1)

if $G == 0$ **then**

// no reconciliation is necessary

`addTriple(t)`

initialise G'_0 where $V' = V$ and $E' = E$

$G'_0 \leftarrow G'_0 \cup \{ \langle s, f_0(s) = 0 \rangle, \langle \langle s, o \rangle, co_0(s, o) = 0 \rangle \}$

return

end if

```

if  $s \in V'$  then
     $f_i(s) = f_{i-1}(s) + 1$ 
    if  $\langle s, o \rangle \in E'$  then
         $co_i(s, o) = co_{i-1}(s, o) + 1$ 
    else
         $co_i(s, o) = 0$ 
    end if
else
     $f_i^s = 0$ 
end if
 $G'_i \leftarrow G'_{i-1} \cup \{ \langle s, f_i(s) \rangle, \langle \langle s, o \rangle, co_i(s, o) \rangle \}$ 

```

addTriple(t)

```

reconcile( $t$ ) against  $G'_i$ 
if new triples are generated then
    new addEvent.fire()
end if

```

Algoritmo 3 reconcile(t) $t = \langle s, p, o \rangle$

Require: α and β as predefined thresholds

$S \leftarrow \{ v \in V' \mid f_i(v) > \alpha \}$

for every $v' \in S$ **do**

doReconciliation(s, v')

end for

for every $v'' \in V' / S$ **do**

if $\exists v' \in S. \langle v', v'' \rangle \in E' \wedge co_i(v', v'') > \beta$ **then**

doReconciliation(s, v'')

else

record(v'', t)

end if

end for

actualizar los metadatos de G'

5

El algoritmo 3 es un algoritmo ejemplar para seleccionar un subconjunto inicial de nodos con el que se lleva a cabo procesado de reconciliación con un nodo particular, y el registro de una indicación del nodo particular en los nodos que no son parte del subconjunto inicial, de modo que se pueden reconciliar con el nodo particular a demanda en el futuro (es decir, el procesado de reconciliación se retarda).

10

La figura 4 ilustra un flujo de control correspondiente al algoritmo 2, y al procesado que tiene lugar como parte de S105 de la figura 3. Es decir, la figura 4 ilustra un flujo de control para realizar procesado de reconciliación para un nodo particular (por ejemplo, el nodo sujeto de una nueva tripla) con el gráfico, y actualizar el gráfico extendido. En S200, se recuperan los nodos más visitados. El procesado de reconciliación entre el nodo particular y los nodos más visitados se efectúa en S201, mientras que la aparición conjunta de nodos se recupera/calcula en S202, y los nodos contiguos a los nodos más visitados que tienen frecuencias de aparición conjunta con el respectivo nodo más visitado superiores a un umbral predeterminado ('Sí' en S203) también son reconciliados con el nodo particular en S201 como parte del subconjunto inicial. El procesado realizado en S201 se ilustra con más detalle en el diagrama de flujo de la figura 5 y en el algoritmo 3. Una vez que no haya nodos más visitados restantes o contiguos a los nodos más visitados que tengan aparición conjunta con el nodo más visitado superior a un umbral, se comprueba si la generación de nuevas triplas como parte del procesado de reconciliación ha dado lugar a nuevos nodos más

20

visitados, de modo que se repiten los pasos S200 a S203. Esto se conoce como el “efecto en cadena” y continúa hasta que no se añaden nuevas triplas como parte del procesado de reconciliación. El procesado de reconciliación entre el resto de los nodos y el nodo particular se retarda. Como se expone en el algoritmo 2, se registra una indicación del nodo particular en cada uno de los nodos con los que no se realiza procesado de reconciliación como parte de los pasos S200 a S204, de modo que los nodos puedan ser reconciliados incrementalmente después de un retardo.

La base que está detrás de este acercamiento incremental es que la reconciliación solamente se realiza con respecto a los nodos que fueron visitados frecuentemente en el pasado y sus contiguos que están fuertemente correlacionados con ellos (a través de aparición conjunta). La reconciliación de otros nodos se pospone hasta que sean visitados o hasta que se realice explícitamente una petición de reconciliación exhaustiva. Como se representa en el algoritmo 3, un subconjunto de nodos más frecuentemente visitados S se identifica usando los valores de frecuencia.

La reconciliación se realiza frente a este subconjunto inicial de nodos más frecuentemente visitados. Todos los nodos no procesados que están fuertemente conectados a nodos frecuentemente visitados (por encima de un umbral predefinido β) son reconciliados frente al nodo particular, los nodos fuertemente conectados pueden ser parte del subconjunto inicial, pero en algunas realizaciones puede ser que el subconjunto inicial sea solamente los nodos más frecuentemente visitados, y que los nodos fuertemente conectados (los que tienen una relación de aparición conjunta $co(s,o)$ con un nodo más visitado por encima de un umbral) son parte del subconjunto adicional de nodos con respecto al que se retarda el procesado de reconciliación con el nodo particular.

La reconciliación de todos los nodos restantes se pospondrá hasta que sean visitados explícitamente, se realice un acercamiento de reconciliación exhaustiva con respecto al nodo particular, o se lleve a cabo procesado de reconciliación en respuesta a una indicación de que el sistema está inactivo o tiene capacidad de procesado de reserva. Sin embargo, la reconciliación no realizada se registra localmente en cada nodo usando `record()`. Esto permitirá la reconciliación de una pasada cuando sea necesaria o favorable.

La reconciliación `doReconciliation(s,v')` para $v' \in V_k$ consta de dos partes específicas: cálculo de semejanza y agregación de resultados. La figura 5 es un diagrama de flujo que expone el procesado realizado como parte de “realizar reconciliación” del paso S201 de la figura 4. El procesado de reconciliación se formaliza en el algoritmo 3 anterior. El algoritmo 3 y la figura 5 exponen procesado de reconciliación ejemplar en una implementación específica de una realización de la invención. Naturalmente, muchas rutinas alternativas de procesado de reconciliación se pueden realizar en realizaciones de la presente invención. La lista de candidatos de reconciliación se obtiene en el paso S300. Éste es, por ejemplo, el subconjunto inicial de nodos, o todo o parte del subconjunto adicional de nodos. En el paso S301, se calculan las medidas de semejanza, o se realizan cálculos de semejanza. Los cálculos de semejanza pueden ser un conjunto de medidas de semejanza para determinar en qué medida los dos recursos s, v' son equivalentes. Cada cálculo de semejanza produce una puntuación que denota el grado de semejanza entre los dos recursos (o la probabilidad de que el nodo particular s y otro nodo v' sean equivalentes), de modo que $similarity(s; v') = \sigma$, siendo σ la puntuación en $[0, 1]$ donde 0 denota que dos entidades son diferentes y 1 denota que son equivalentes (describen el mismo objeto del mundo real). Los cálculos de semejanza pueden combinarse entonces, por ejemplo, por simple agregación o por media ponderada, para obtener un valor de semejanza.

El cálculo de semejanza se puede basar en diferentes medidas de semejanza, la mayoría implica varias métricas de semejanza en cadena, ampliamente usadas tanto en enlaces de registro como en adaptación de esquemas. La métrica de semejanza en cadena puede incluir la distancia Levenshtein Edit, que mide el número de operaciones de cambio primitivas necesarias para transformar un valor de cadena a otro, y/o la métrica Jaro, diseñada para tomar en cuenta desviaciones ortográficas comunes. Éstas dos son simplemente una muestra de las muchas realizaciones de medidas de semejanza que se puede usar para realizar reconciliación de recursos. En la agregación de resultados, por cada par $(s, v', \text{ puntuación})$, la reconciliación obtiene, por ejemplo, un valor de semejanza agregado equivalente $(s, v_i, \text{ weighted_score})$ como una media ponderada de estas medidas de semejanza. Solamente si el valor de semejanza es superior a un umbral de semejanza elegido (S303), `similarity_th`, el enlace de equivalencia candidato será seleccionado y añadido en el gráfico. `Similarity_th` denota el valor mínimo de semejanza necesario para considerar un par de descripciones de objeto como similares una a otra.

La reconciliación pospuesta se maneja en una o dos formas diferentes:

i) reconciliación “a demanda” y “en nodo” disparada por `readEvent` (consultas); y

ii) reconciliación forzada en tiempo de reposo del sistema. En los casos extremos, es posible que no se realice reconciliación al tiempo de introducción de tripla, es decir, toda la reconciliación se puede posponer al tiempo de reposo del sistema o el punto de consulta explícita.

1) *Consultar gráfico con peticiones de reconciliación pendientes:*

El acercamiento de reconciliación incremental que se usa en realizaciones de la presente invención incluye

posponer o retardar el procesado de reconciliación entre un nodo particular con el que se ha determinado que se deberá realizar reconciliación con los nodos existentes en el gráfico, y al menos algunos de los nodos en el gráfico, hasta un punto de tiempo en que sea realmente necesario o favorable. Como se ha explicado anteriormente, puede ser que el nodo particular sea el nodo sujeto de una nueva tripla insertada en el gráfico. Al tiempo de la introducción, en el nodo particular puede haberse realizado solamente procesado de reconciliación entre sí mismo y los nodos/recursos más frecuentemente visitados y sus contiguos fuertemente correlacionados. La reconciliación frente a todos los nodos restantes se retarda para consideración "a demanda".

Como un mecanismo por el que el procesado de reconciliación retardado puede ser identificado y programado, estas peticiones pendientes pueden almacenarse en una o dos posiciones diferentes. Efectivamente, el espacio de almacenamiento adicional requerido para este mecanismo es un compromiso de la eficiencia de tiempo y el rendimiento del procesado.

En primer lugar, las triplas que inician un proceso de reconciliación de datos se almacenan en cada nodo no visitado del gráfico extendido como metadatos de nodo (véase la definición 4 y la definición 6). Almacenando peticiones pendientes localmente en el nodo (en la lista L_i), la reconciliación adicional puede proseguir localmente en cada nodo y a demanda, es decir, el hecho de que haya procesado retardado con respecto a un nodo puede ser conocido leyendo los metadatos almacenados para dicho nodo, y el procesado retardado puede ser programado consiguientemente.

En segundo lugar, el sistema mantiene una lista de triplas nuevamente añadidas que todavía no están completamente reconciliadas (donde una tripla o nodo completamente reconciliado es uno para el que ha sido realizado el procesado de reconciliación con cada uno de los otros nodos en el gráfico) al gráfico. Esta lista se denomina la cola (o cache) de triplas no reconciliadas, $Q = [t_0, \dots, t_m]$.

La figura 6 es un diagrama de flujo que expone el procesado realizado a la recepción de una consulta en el gráfico, con el fin de asegurar que el procesado de reconciliación con respecto a los nodos sea recuperado en respuesta a una consulta realizada frente a todos los otros nodos en el gráfico.

Al consultar, una consulta (q) es verificada primero frente a los elementos en Q , en los pasos S401 y S402.

La condición a cumplir para que se considere que la consulta incluye un elemento en Q puede tomar la forma:

$$(\exists t_j \in Q.t_j = q \vee similarity(t_j, q) \geq \delta)$$

donde δ es un umbral numérico predefinido. Esencialmente, si la consulta es igual a un elemento en Q o es significativamente similar a un elemento en Q (donde la semejanza umbral viene dada por δ), entonces la reconciliación de datos del nodo en cuestión se realiza frente a todo el gráfico. La provisión de un umbral de semejanza proporciona flexibilidad al verificar si Q contiene la consulta deseada.

Si Q contiene la consulta (es decir, si se cumple la condición anterior), 'SÍ' en S402, el flujo pasa a S403 y la reconciliación de datos del nodo consultado que es un elemento de Q , t_j , se refuerza frente a todo el gráfico. Obsérvese que esta 'reconciliación exhaustiva' define efectivamente un 'escenario del peor caso' en términos de la cantidad de procesado de reconciliación que debe realizarse a la recepción de una consulta. Sin embargo, en la mayoría de las implementaciones se espera que el tamaño de Q sea mucho menor en comparación con el gráfico. Por lo tanto, la reconciliación exhaustiva forzada del gráfico no deberá suceder frecuentemente. Además, en el tiempo de reposo del sistema las triplas no reconciliadas en cola son quitadas y procesadas, reduciendo más la posibilidad de que una consulta se encuentre en el 'escenario del peor caso'. Además, el caso es que la reconciliación exhaustiva se requiere en este punto debido a la identificación del nodo en una consulta, por lo tanto el procesado es eficiente en términos del hecho de que la reconciliación es con respecto a un nodo al que se está accediendo. Después de la reconciliación exhaustiva en S403 (se expone un proceso de reconciliación exhaustiva ejemplar con más detalle en la figura 7), el flujo pasa a S404, y se evalúa la consulta y se leen los nodos.

Si la respuesta es 'NO' en S402, y no hay solapamiento entre la consulta q y la cola Q , el flujo pasa a S405 y se evalúa frente al gráfico extendido. En los pasos S405 y S406 se lleva a cabo una comprobación para identificar si alguno de los nodos consultados tiene indicaciones "en nodo" (identificadas en la cola L_i almacenada en los metadatos) de nodos con los que tienen procesado de reconciliación pendiente/retardado, tan pronto como se encuentra un nodo no reconciliado (nodos en el gráfico extendido con reconciliación pendiente), se realiza procesado de reconciliación de datos "en nodo" en S407. Un procesado ejemplar para el procesado de reconciliación de datos "en nodo" se expone en la figura 8.

En el paso S408, durante la evaluación de consulta, los metadatos del gráfico extendido son actualizados en términos de frecuencia, aparición conjunta y peticiones de reconciliación pendiente en cache tanto de nodo local como global.

Como se puede ver en las figuras 8 y 9, el procesado de reconciliación que debe realizarse en cada caso es en gran parte el mismo que el procesado de reconciliación expuesto en la figura 5, y se han usado los mismos números para pasos directamente equivalentes. La diferencia está simplemente en los nodos candidato con los que se ha de realizar procesado de reconciliación con el nodo particular.

2) Reconciliación en tiempo de reposo: aparte de la reconciliación de datos “a demanda” y “en nodo” forzada por consultas, el acercamiento incremental también permite una mejor utilización de los recursos informáticos. Supervisando todo el sistema, es posible identificar intervalos de tiempo cuando los recursos informáticos están en reposo, o en un estado operativo que se determina que tiene suficiente capacidad de procesado para añadir procesado de reconciliación a la carga de trabajo del recurso informático. La reconciliación pendiente puede ser identificada, programada y realizada frente a todo el gráfico si es necesario y apropiado o incrementalmente usando de nuevo los valores de frecuencia y aparición conjunta. Obsérvese que la reconciliación también produce la actualización de los metadatos de gráfico extendido. La reconciliación en tiempo de reposo puede ser una o ambas de reconciliación “en nodo” y reconciliación exhaustiva.

Implementar un acercamiento activado por evento tiene la característica importante de que permite la existencia de varias tareas de semejanza a ejecutar simultáneamente en nodos de Internet diferentes y distribuidos. éste se diferencia de otros acercamientos. El uso de eventos es el principal vehículo para organizar intercomunicación de componentes. La paralelización de la operación semántica se logra a través de una segmentación de datos bidimensional.

En primer lugar, el gráfico original de datos es distribuido por múltiples almacenamientos físicos, lo que permite una segmentación de datos ‘vertical’ cuando se realiza la reconciliación. Diferentes tareas de semejanza pueden ser realizadas por servidores remotos que estén enlazados a través de una red de comunicaciones y tales tareas pueden ejecutarse en paralelo. La distribución y la paralelización son características de un sistema activado por evento que puede ayudar a acelerar el tiempo de ejecución de todo el proceso de reconciliación. El procesado de reconciliación puede ser realizado por un recurso de cálculo disparado por un evento de lectura o un evento de adición con respecto a un nodo particular que todavía no ha sido reconciliado con todos los nodos en dicho recurso de cálculo. Más de un recurso de cálculo puede realizar tal procesado de reconciliación con respecto al mismo nodo particular en un tiempo.

La figura 9 es una ilustración de la paralelización del procesado de reconciliación a través de un número de servidores. El procesado de reconciliación es disparado por un evento, que puede ser la introducción de una nueva tripla al almacenamiento de datos subyacente, en cuyo caso el sujeto de la tripla es el “particular”. El rendimiento de cálculos de semejanza y el procesado requerido para obtener un valor de semejanza en base a los cálculos puede ser realizado en cada servidor entre el nodo particular y los nodos de una lista de candidatos seleccionada (por ejemplo, el subconjunto inicial, o en el caso de procesado exhaustivo, todo el gráfico) que están almacenados en dicho servidor. En ese sentido, el procesado de reconciliación se descompone en un número de tareas de semejanza (tareas de semejanza 1 a n) que se realizan en paralelo en los servidores individuales. Los resultados se agregan entonces y el gráfico se actualiza con enlaces de equivalencia donde sea apropiado, lo que puede dar origen a otro “evento” que dispare más procesado de reconciliación.

En segundo lugar, el gráfico extendido ofrece una mayor capacidad de paralelización a través de una capa abstracta y dinámica de división de datos. Usando los valores de frecuencia y aparición conjunta almacenados como metadatos en el gráfico extendido G’ como un discriminador, es posible imponer una segmentación de datos ortogonal a la física subyacente adquirida a través de almacenamiento de datos distribuido. Los datos en el gráfico son troceados en dados después efectivamente en elementos más pequeños para procesado de reconciliación, definiendo cada dado de datos los límites dentro de los que una rutina de procesado de reconciliación está confinada (como se ilustra en la figura 10). Esta segunda segmentación es deseable porque, aunque el gráfico esté almacenado por múltiples máquinas, el tamaño de datos en una máquina individual puede ser potencialmente muy grande denegando una acción de reconciliación exhaustiva en tiempo real. La probabilidad de aparición conjunta entre dos nodos arbitrarios puede ser usada para dividir el gráfico en un gráfico secundario de nodos frecuentemente visitados (un subconjunto inicial) y otro conteniendo menos nodos visitados (un subconjunto adicional). Con respecto a éste último, el procesado de reconciliación se retarda o pospone hasta que se realice una operación de lectura directa, o hasta el tiempo de reposo del sistema. Por lo tanto, se habilita un acercamiento incremental que puede paralelizar potencialmente el procesado de reconciliación semántica de datos. Superponiéndose a la segmentación de almacenamiento físico ya en posición, esta segmentación basada en el uso mejora más el procesado paralelo de reconciliación de datos.

En cualquiera de los aspectos anteriores, las varias características pueden implementarse en hardware, o como módulos de software que se ejecutan en uno o varios procesadores. Las características de un aspecto pueden aplicarse a cualquiera de los otros aspectos.

La invención también proporciona un programa de ordenador o un producto de programa de ordenador para llevar a la práctica cualquiera de los métodos aquí descritos, y un medio legible por ordenador que almacena un programa para llevar a la práctica cualquiera de los métodos aquí descritos. Un programa de ordenador que realiza la

invención puede estar almacenado en un medio legible por ordenador, o podría estar, por ejemplo, en forma de una señal tal como una señal de datos descargable proporcionada desde una página de Internet, o podría estar en cualquier otra forma.

REIVINDICACIONES

1. Un método para realizar reconciliación de recursos representados en un gráfico de datos como nodos; incluyendo el método:
- 5 seleccionar un subconjunto inicial de los nodos del gráfico para los que se ha de realizar procesado de reconciliación con un nodo particular;
- realizar procesado de reconciliación entre el nodo particular y cada uno del subconjunto inicial de nodos;
- 10 retardar el procesado de reconciliación entre el nodo particular y otro subconjunto de los nodos del gráfico para el que se ha de realizar reconciliación con el nodo particular, donde el procesado de reconciliación del subconjunto adicional de los nodos se retarda y realiza durante intervalos de tiempo cuando la supervisión de los recursos informáticos responsables del procesado de reconciliación indica que dichos recursos informáticos estarían en reposo en caso contrario.
- 15
2. Un método según la reivindicación 1, donde
- seleccionar el subconjunto inicial incluye determinar qué nodos tienen el número más alto de accesos de lectura realizados en ellos en operaciones de representación gráfica, e incluyendo los nodos en el subconjunto inicial como los nodos más visitados.
- 20
3. Un método según la reivindicación 2, donde
- seleccionar el subconjunto inicial también incluye determinar qué nodos tienen la probabilidad más alta de que en ellos se haga un acceso de lectura como el nodo siguiente después de un nodo de los nodos más visitados en una operación de representación gráfica.
- 25
4. Un método según cualquiera de las reivindicaciones precedentes, donde
- 30 el subconjunto adicional de los nodos son todos los nodos en el gráfico que no son elementos del subconjunto inicial.
5. Un método según cualquiera de las reivindicaciones precedentes, incluyendo además:
- 35 añadir una identificación del nodo particular a una lista;
- mantener la identificación del nodo particular en la lista hasta que se haya realizado el procesado con el subconjunto adicional de los nodos del gráfico, después de lo que la identificación del nodo particular se quita de la lista; y
- 40 cuando se recibe una consulta al gráfico que identifica uno o más nodos a leer, verificar cuál o cuáles nodos a leer son identificados en la lista, y para cada uno del único o varios nodos a leer que es identificado en la lista, realizar procesado de reconciliación con cada nodo del gráfico.
- 45
6. Un método según cualquiera de las reivindicaciones precedentes, incluyendo además:
- por cada nodo del gráfico no incluido en el subconjunto inicial con el que se lleva a cabo procesado de reconciliación,
- 50 añadir una identificación del nodo particular a una lista de nodos no reconciliados en los metadatos asociados, y mantener dicha identificación en dicha lista hasta que se lleve a cabo procesado de reconciliación con el nodo particular, después de lo que la identificación del nodo particular se quita de dicha lista; y
- cuando se recibe una consulta al gráfico que identifica uno o más nodos a leer, por cada uno de los nodos a leer, realizar procesado de reconciliación entre dicho nodo y cada uno de los nodos identificados en la lista de nodos no reconciliados en los metadatos asociados.
- 55
7. Un método según cualquiera de las reivindicaciones precedentes, donde el gráfico se almacena en una red distribuida de recursos informáticos.
- 60
8. Un método según cualquiera de las reivindicaciones precedentes, donde el procesado de reconciliación del nodo particular con otro nodo incluye:
- 65 obtener un valor de semejanza agregando los valores resultantes de realizar una pluralidad de diferentes cálculos de semejanza, u obtener un solo valor resultante de realizar un cálculo de semejanza, donde un cálculo de semejanza es un cálculo que genera un valor que representa la probabilidad de que el nodo particular y el otro nodo sean

semánticamente equivalentes; y

si el valor de semejanza excede de un umbral de semejanza predeterminado, añadir un enlace de equivalencia al gráfico que denota una relación de equivalencia entre el nodo particular y el otro nodo.

5 9. Un método según las reivindicaciones 7 y 8, donde
el procesado de reconciliación se realiza simultáneamente en más de un recurso de cálculo dentro de la red distribuida de recursos informáticos, entre el nodo particular y los nodos almacenados en dicho recurso de
10 computación y pertenecientes al subconjunto de nodos para el que se está realizando procesado de reconciliación con el nodo particular.

10. Un método según cualquiera de las reivindicaciones precedentes, incluyendo además

15 almacenar, por cada nodo, como metadatos asociados, un valor que representa los números de accesos de lectura hechos al nodo en operaciones de representación gráfica, y un valor que representa el número de veces que se leyó cada nodo al que el nodo está enlazado en el gráfico como el nodo siguiente en una operación de representación gráfica.

20 11. Un método según cualquiera de las reivindicaciones precedentes, donde los datos en el gráfico son codificados como triplas incluyendo cada una un sujeto, un predicado, y un objeto, y los nodos del gráfico son los sujetos y objetos de las triplas, y el predicado de una tripla denota un enlace entre el sujeto y el objeto de la tripla.

25 12. Un controlador de base de datos (21) para realizar reconciliación de recursos representados en un gráfico de datos como nodos; incluyendo el controlador de base de datos (21):

un módulo de selección configurado para seleccionar un subconjunto inicial de los nodos del gráfico para los que se ha de realizar procesado de reconciliación con un nodo particular;

30 un módulo de procesado de reconciliación configurado para realizar procesado de reconciliación entre el nodo particular y cada uno del subconjunto inicial de nodos; y

un módulo de control de procesado de reconciliación configurado para retardar el procesado de reconciliación de otro subconjunto de los nodos del gráfico para el que se ha de realizar reconciliación con el nodo particular, donde el
35 procesado de reconciliación del subconjunto adicional de los nodos se retarda y realiza durante intervalos de tiempo cuando la supervisión de recursos informáticos responsables del procesado de reconciliación indica que dichos recursos informáticos estarían en reposo en caso contrario.

40 13. Un sistema de almacenamiento de datos para almacenar un gráfico de datos en el que los recursos se representan como nodos del gráfico, incluyendo el sistema de almacenamiento de datos:

una pluralidad de unidades de almacenamiento (23) configurada cada una para almacenar un segmento de datos del gráfico de datos; y un controlador de base de datos (21) según la reivindicación 12.

45 14. Un programa de ordenador que, cuando es ejecutado por un ordenador, hace que el ordenador realice el método según cualquiera de las reivindicaciones 1 a 11, o hace que el ordenador funcione como el controlador de base de datos según la reivindicación 12.

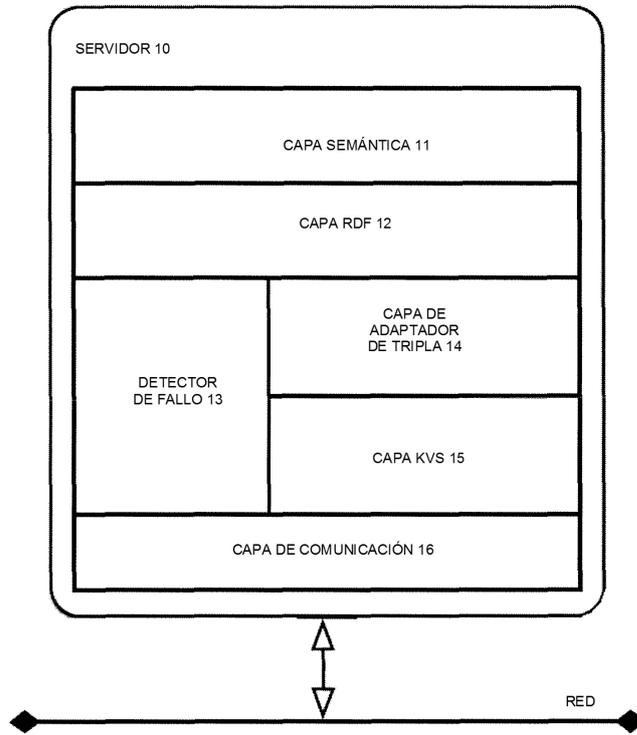


FIGURA 1

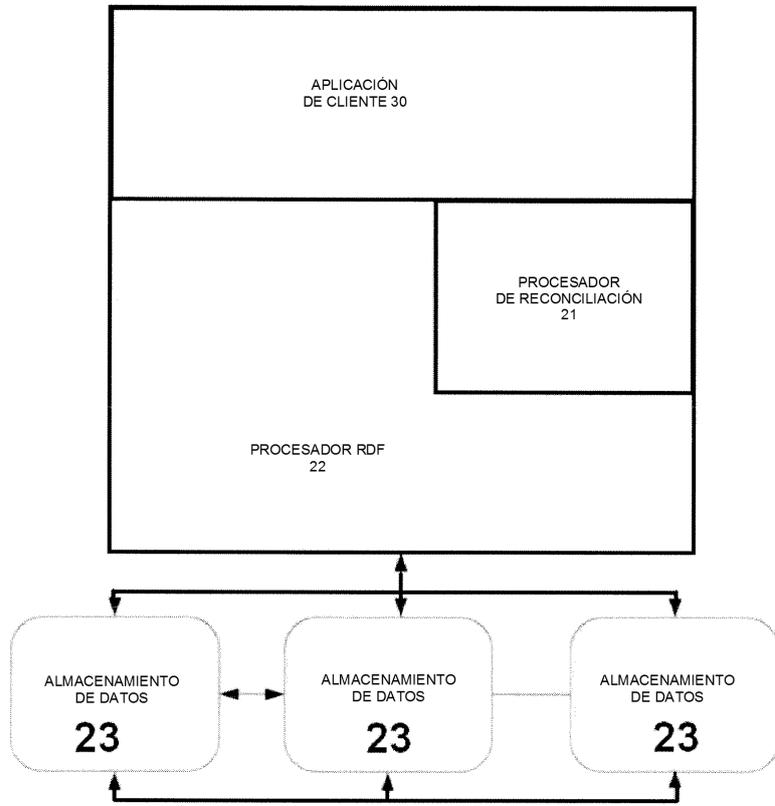


FIGURA 2

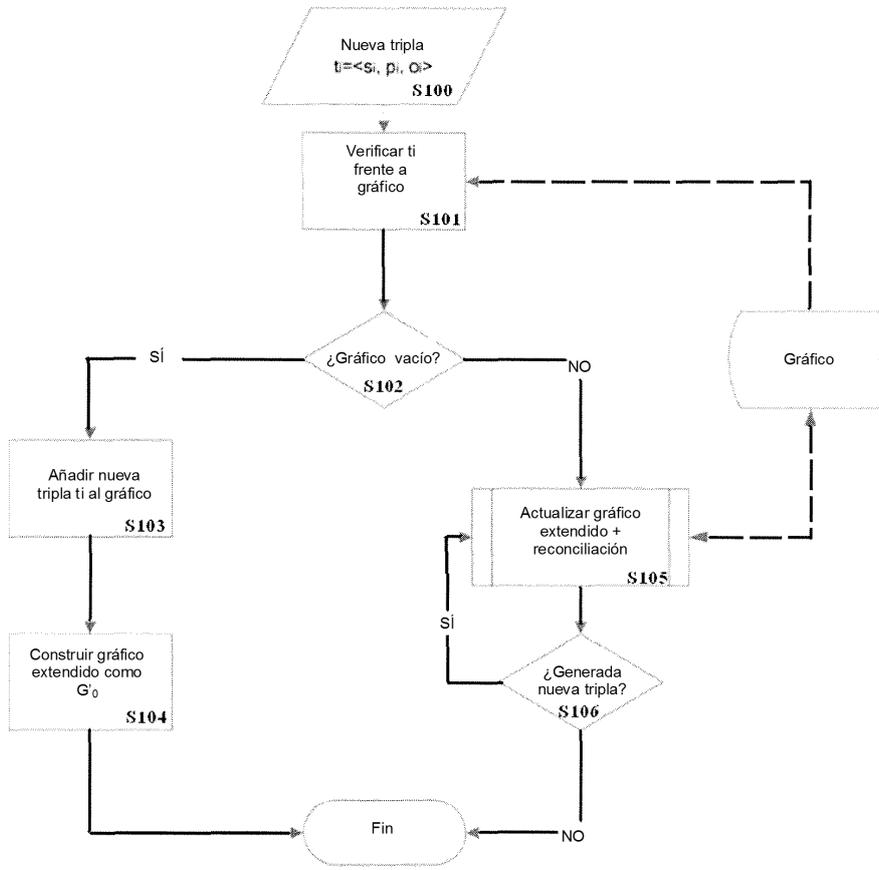


Figura 3: Añadir nuevas triplas

FIGURA 3

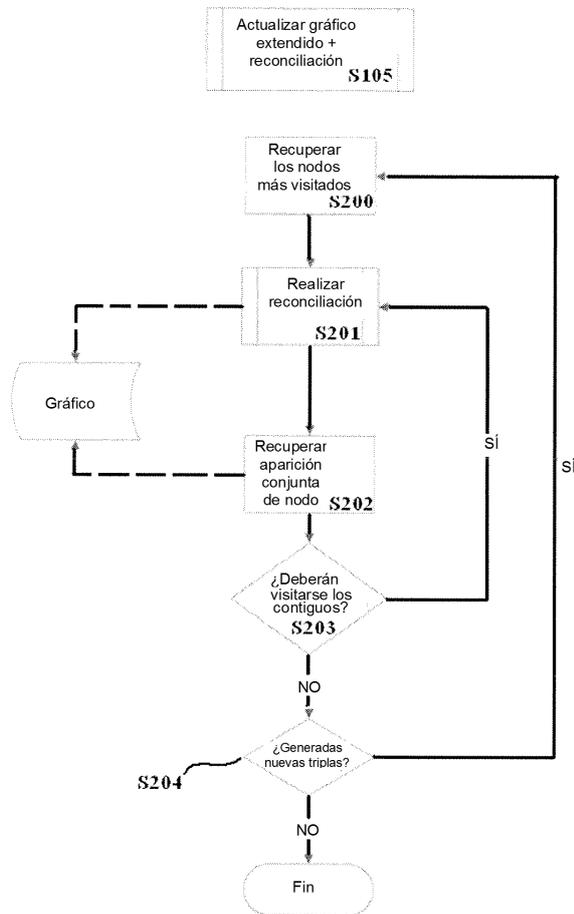


Figura 4: Construir/actualizar gráfico extendido

FIGURA 4

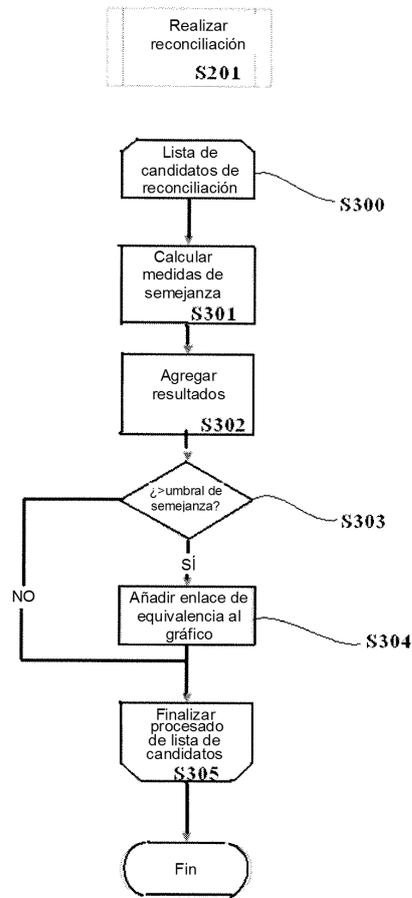


Figura 5: Proceso de reconciliación

FIGURA 5

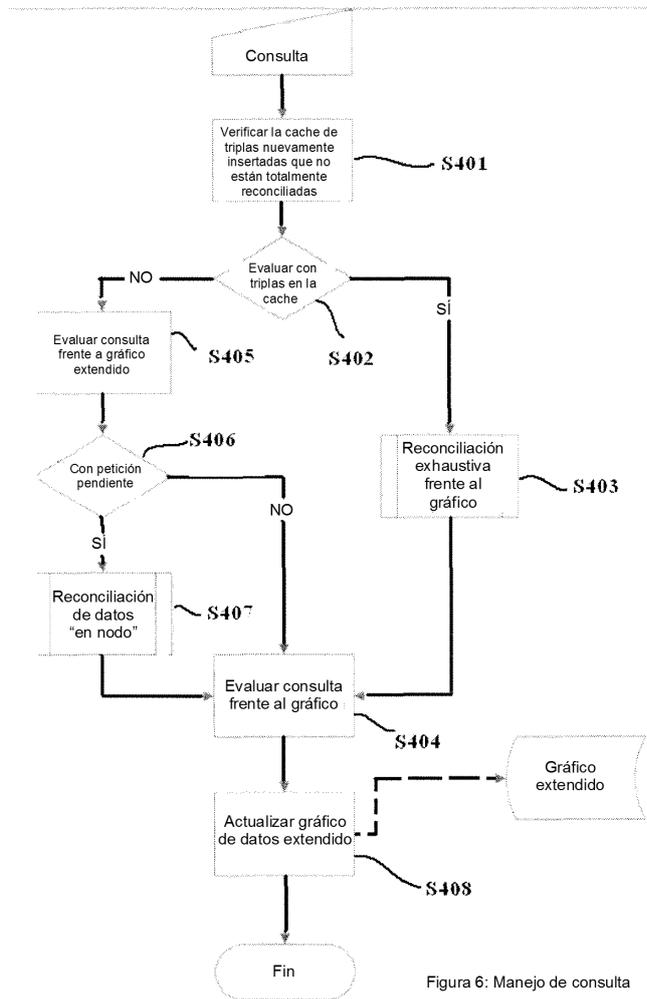


Figura 6: Manejo de consulta

FIGURA 6

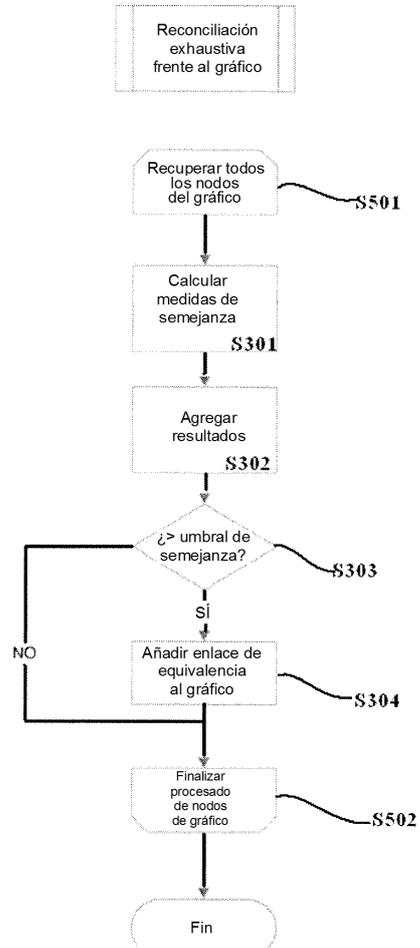


Figura 7: Reconciliación exhaustiva

FIGURA 7

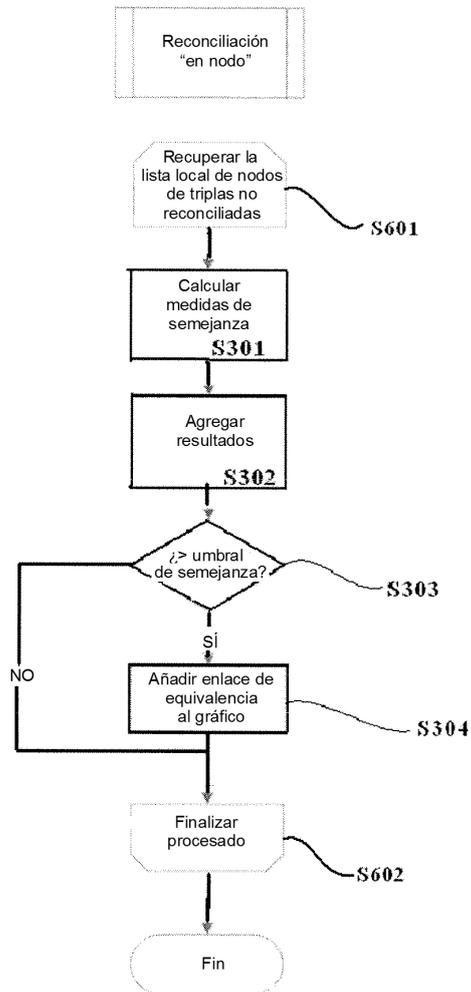


Figura 8: Reconciliación "en nodo"

FIGURA 8

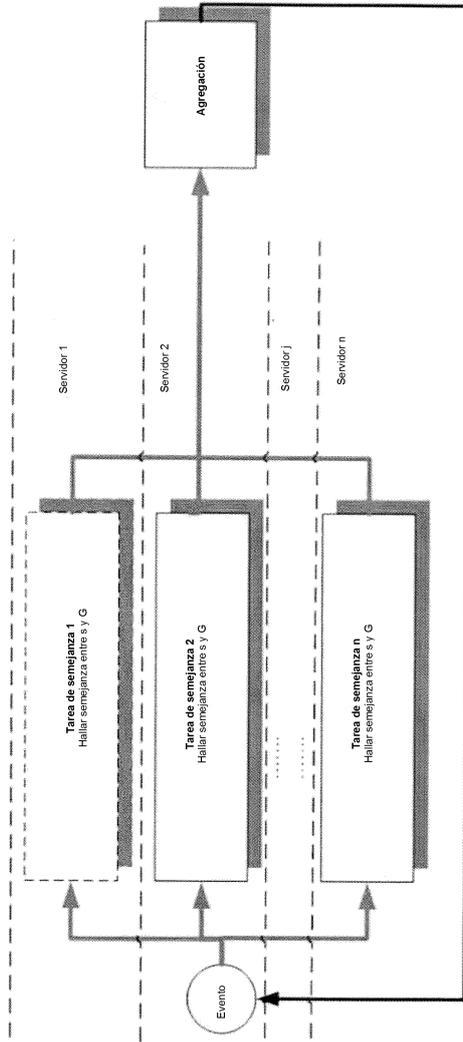


Figura 9: Distribución y paralelización de tareas de semejanza

FIGURA 9

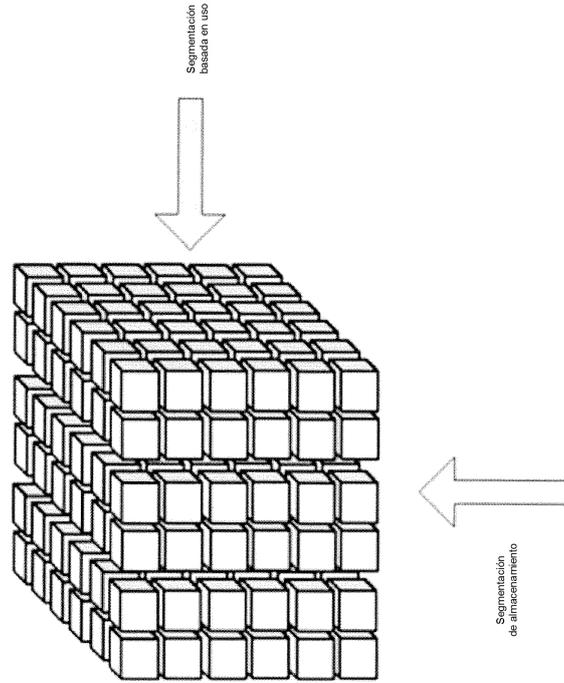


Figura 10: Segmentación bidimensional de datos

FIGURA 10