(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 845 299**

(51) Int. CI.:

H04L 12/709 (2013.01)
H04L 29/06 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **18.09.2013 PCT/US2013/060409**

(87) Fecha y número de publicación internacional: **27.03.2014 WO14047182**

(96) Fecha de presentación y número de la solicitud europea: **18.09.2013 E 13839088 (5)**

(97) Fecha y número de publicación de la concesión europea: **28.10.2020 EP 2898638**

(54) Título: **Transmisión de datos en flujo de alto rendimiento**

(30) Prioridad:

21.09.2012 US 201261704302 P
13.03.2013 US 201361778872 P
06.06.2013 US 201361832075 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
26.07.2021

(73) Titular/es:

NYSE GROUP, INC. (100.0%)
11 Wall Street
New York, NY 10005, US

(72) Inventor/es:

WERR, EMILE

(74) Agente/Representante:

CURELL SUÑOL, S.L.P.

DESCRIPCIÓN

Transmisión de datos en flujo de alto rendimiento

Campo técnico

El campo se refiere, en general, al procesado de datos y, más específicamente, a la transferencia de datos sobre entornos distribuidos.

Antecedentes

Aproximadamente se generan 2.5 trillones de bytes de datos a nivel mundial cada día. Además, se estima que el 90% de los datos del mundo se ha producido solo en los dos últimos años.

El término “big data” se refiere a recopilaciones de grandes conjuntos de datos complejos. La gestión de una recopilación enorme de datos presenta muchos desafíos que incluyen capturar, almacenar, buscar, transformar, transferir y analizar dichos datos. En particular, las herramientas existentes de procesado de datos no tienen la capacidad de manipular y transportar cantidades masivas de datos de manera suficientemente rápida para satisfacer los requisitos comerciales.

El documento US 2007/185938 A1 divulga la ejecución de operaciones de gestión de datos sobre datos duplicados en una red informática, en donde se puede implementar paralelismo entre múltiples hilos cuando se lleva a cabo una duplicación de datos. El documento US 2010/082543 A1 divulga un sistema para migrar datos dentro de una red de área de almacenamiento. Se crea un plan de migración para mover datos almacenados en la red de área de almacenamiento, en donde cada elemento de datos debe moverse desde una ubicación de origen hasta una ubicación de destino de acuerdo con un mapeo entre ellas y en donde se pueden ejecutar operaciones en paralelo.

Por consiguiente, existe una necesidad de una solución de alto rendimiento para procesar, transformar y distribuir de manera rápida grandes cantidades de datos de una forma tal que satisfaga las demandas de los clientes, las necesidades comerciales y los acuerdos a nivel de servicios.

Esta solución la aportan un método implementado por ordenador, un sistema y un soporte legible por ordenador que comprenden las características de las reivindicaciones independientes. En las reivindicaciones dependientes se definen formas de realización preferidas de la invención.

Sumario

Las formas de realización se refieren, en general, a la transmisión de datos en flujo de alto rendimiento. En una de las formas de realización, un procesador recibe un mapeo de datos que describe una asociación entre uno o más campos de una ubicación de almacenamiento de datos de una fuente de datos y uno o más campos de una ubicación de almacenamiento de datos de un destino objetivo. El procesador genera un plan de ejecución de transferencia de datos a partir del mapeo de datos para transferir datos desde la fuente de datos hasta el destino objetivo donde el plan de ejecución de transferencia de datos comprende un grado determinado de paralelismo a usar cuando se transfieren los datos. El procesador también transfiere los datos desde la ubicación de almacenamiento de la fuente de datos hasta la ubicación de almacenamiento de datos del destino objetivo usando el plan generado de ejecución de transferencia de datos.

En otra forma de realización, un sistema incluye una memoria y un procesador acoplado a la memoria para proporcionar una transmisión de datos en flujo de alto rendimiento. El sistema recibe un mapeo de datos que describe una asociación entre uno o más campos de una ubicación de almacenamiento de datos de una fuente de datos y uno o más campos de una ubicación de almacenamiento de datos de un destino objetivo. El sistema genera un plan de ejecución de transferencia de datos a partir del mapeo de datos para transferir datos desde la fuente de datos hasta el destino objetivo donde el plan de ejecución de transferencia de datos comprende un grado determinado de paralelismo a usar cuando se transfieren los datos. El sistema también transfiere los datos desde la ubicación de almacenamiento de la fuente de datos hasta la ubicación de almacenamiento de datos del destino objetivo usando el plan generado de ejecución de transferencia de datos.

En una forma de realización adicional, un soporte legible por ordenador tiene instrucciones que, cuando son ejecutadas por un procesador, provocan que el procesador lleve a cabo operaciones. Las instrucciones incluyen código de programa legible por ordenador, configurado para provocar que el procesador reciba un mapeo de datos que describe una asociación entre uno o más campos de una ubicación de almacenamiento de datos de una fuente de datos y uno o más campos de una ubicación de almacenamiento de datos de un destino objetivo. Las instrucciones incluyen, también, un código legible por ordenador, configurado para provocar que el procesador genere un plan de ejecución de transferencia de datos a partir del mapeo de datos con el fin de transferir datos desde la fuente de datos hasta el destino objetivo donde el plan de ejecución de transferencia de datos comprende un grado determinado de paralelismo a usar cuando se transfieren los datos. Las instrucciones incluyen, además,

un código legible por ordenador configurado para provocar que el procesador transfiera los datos desde la ubicación de almacenamiento de la fuente de datos hasta la ubicación de almacenamiento de datos del destino objetivo usando el plan generado de ejecución de transferencia de datos.

En la presente memoria, se describen de forma detallada otras formas de realización, características y ventajas de la presente invención, así como la estructura y el funcionamiento de las diversas formas de realización de la presente invención.

Breve descripción de los dibujos

La presente invención se ilustra a título de ejemplo, y no a título limitativo, y se pondrá de manifiesto al considerar la siguiente descripción detallada, considerada en combinación con los dibujos adjuntos, en los que los caracteres de referencia iguales se refieren a las mismas partes en todos ellos, y en los cuales:

la figura 1 ilustra un diagrama de bloques de una arquitectura de un sistema de transmisión de datos en flujo de alto rendimiento, de acuerdo con varias formas de realización de la presente invención.

la figura 2 es un diagrama de flujo que ilustra la transmisión de datos en flujo de alto rendimiento, de acuerdo con una forma de realización de la presente invención.

la figura 3 es un diagrama de flujo que ilustra aspectos adicionales de la transmisión de datos en flujo de alto rendimiento, de acuerdo con una forma de realización de la presente invención.

la figura 4 es un diagrama de bloques de un sistema de ordenador ejemplificativo que puede llevar a cabo una o más de las operaciones descritas en la presente memoria.

Descripción detallada

La presente invención va dirigida a sistemas, métodos y productos de programa de ordenador para la transmisión de datos en flujo de alto rendimiento. En una de las formas de realización, un módulo transmisor de datos en flujo de alto rendimiento es un sistema de transferencia de datos de alta velocidad que lleva a cabo una transferencia rápida de grandes conjuntos de datos entre entornos distribuidos. Por ejemplo, un módulo de transmisión de datos en flujo de alto rendimiento proporciona un transporte de datos rápido y fiable entre medios de almacenamiento de datos distribuidos en y entre organizaciones en cualquier lugar del mundo.

A diferencia de las herramientas tradicionales de transferencia de archivos (por ejemplo, FTP, SFTP, RCP, etcétera), un módulo transmisor de datos en flujo de alto rendimiento es singular en la medida en la que admite el movimiento de archivos de todos los tipos, se integra con Hadoop, se interconecta con cualquier tecnología de base de datos/almacenamiento de datos, incluye un repositorio de metadatos para la configuración de mapeos de fuente-a-objetivo, proporciona seguridad y derechos de usuario detallados para el acceso a datos y las operaciones de datos, incluye una interfaz gráfica de usuario (GUI) para usuarios finales y proporciona una interfaz de programación de aplicaciones (API) para la integración en sistemas de fondo (*back-end*).

La figura 1 ilustra un diagrama de bloques de una arquitectura de sistema de transmisión de datos en flujo de alto rendimiento 100, de acuerdo con varias formas de realización de la presente invención.

La arquitectura de sistema de transmisión de datos en flujo de alto rendimiento 100 incluye clientes 101A y 101B, un primer conjunto de fuentes/objetivos de datos 102A-102C, un segundo conjunto de fuentes/objetivos de datos 104A-140C, unas redes 106A y 106B, un sistema de transmisión de datos en flujo 108 y un catálogo de metadatos 110.

La arquitectura de sistema de transmisión de datos en flujo de alto rendimiento 100 incluye uno o más sistemas de ordenador físicos y/o virtuales conectados a una red, tal como las redes 106A y 106B. La red puede ser, por ejemplo, una red pública (por ejemplo, Internet), una red privada (por ejemplo, una red de área local (LAN), una red de área extensa (WAN)), un sistema de archivos de alta definición (HDFS), una red de área de almacenamiento (SAN), unos medios de almacenamiento propios de la red (NAS), unas comunicaciones entre procesos (IPC) o cualquier combinación de los mismos.

Los sistemas de ordenador pueden incluir ordenadores personales (PC), ordenadores portátiles, teléfonos móviles, ordenadores de tipo tableta, o cualquier otro dispositivo informático. Los sistemas de ordenador pueden ejecutar un sistema operativo (OS) que gestiona *hardware* y *software*. Los sistemas de ordenador también pueden incluir una o más máquinas de servidor. Una máquina de servidor puede ser un servidor instalable en bastidor, un ordenador rúter, un ordenador personal, un asistente digital portátil, un teléfono móvil, un ordenador portátil, un ordenador de tipo tableta, una cámara, una videocámara, un *netbook*, un ordenador de sobremesa, un centro de medios o cualquier combinación de los mismos. En un ejemplo, los clientes 101A y 101B, las fuentes/objetivos de datos 102A -102C, las fuentes/objetivos de datos 104A-104C, y el sistema de transmisión de datos en flujo 108 se

proporcionan, cada uno de ellos, usando uno o más sistemas de ordenador.

La arquitectura de sistema de transmisión de datos en flujo de alto rendimiento 100 también puede incluir unos medios de almacenamiento persistente de datos, tales como un servidor de archivos o medios de almacenamiento en red, capaces de almacenar diversos tipos de datos. En algunas formas de realización, los medios de almacenamiento de datos podrían incluir otro u otros tipos de medios de almacenamiento persistente, tales como una base de datos orientada a objetos, una base de datos relacional, una base de datos en memoria y otros. En un ejemplo, el catálogo de metadatos 110 puede residir dentro de unos únicos medios de almacenamiento de datos o sobre múltiples medios diferentes de almacenamiento de datos, lógicos/físicos.

Los clientes 101A y 101B pueden ser aplicaciones, utilidades, herramientas u otro *software* automatizados o controlados por el usuario, conectados al sistema de transmisión de datos en flujo 108 y que se comunican con el mismo. Los clientes 101A y 101B también pueden ser sistemas de ordenador que generan y envían llamadas basadas en interfaces de programación de aplicaciones (API) o en otros servicios al sistema de transmisión de datos en flujo 108, por ejemplo, para transferir datos desde una fuente de datos (por ejemplo, 102A, 102B, 102C) hasta un destino objetivo (por ejemplo, 104A, 104B, 104C), o viceversa cuando proceda.

En un ejemplo, el movimiento de datos entre dos sistemas de ordenador diferentes lo orquesta el sistema de transmisión de datos en flujo 108. Un sistema de origen contiene datos que se van a transferir a un sistema objetivo. Los datos que se van a transferir se basan en un mapeo entre el sistema de origen y el sistema objetivo. El sistema de transmisión de datos en flujo 108 permite el mapeo, la orquestación y la transmisión en flujo de datos hacia y desde varios tipos de soluciones diferentes de almacenamiento de datos que incluyen, aunque sin carácter limitativo, medios de almacenamiento propios de la red (NAS), tecnologías de base de datos, sistemas de archivos, aparatos de almacenamiento de datos, etcétera.

Por ejemplo, se pueden transmitir en flujo datos entre cualquier tipo de recursos similares o diferentes de almacenamiento de datos (por ejemplo, de sistema de archivos a sistema de archivos, de base de datos a base de datos, de sistema de archivos a base de datos, de base de datos a sistema de archivos, de aparato a aparato, de sistema de archivos a aparato, de aparato a sistema de archivos, de aparato a base de datos, de base de datos a aparato, etcétera). Además, los datos se pueden transmitir en flujo de un punto a otro, se pueden fusionar en un único punto, se pueden dividir, filtrar, agregar, transformar y/o transmitir en flujo hacia uno o más destinos diferentes de forma independiente y/o simultánea.

Un primer conjunto de fuentes/objetivos de datos 102A-102C representa en general sistemas de ordenador que almacenan datos, por ejemplo, en una primera ubicación o en nombre de una organización particular. Un segundo conjunto de fuentes/objetivos de datos 104A-104C representa en general sistemas de ordenador que almacenan o almacenarán datos, por ejemplo, en una segunda ubicación o en nombre de una organización diferente. En algunos ejemplos, se pueden transmitir en flujo datos en la misma ubicación física (por ejemplo, centro de datos), dentro de la misma organización y en el mismo sistema de ordenador. Además, un único sistema de ordenador puede comprender muchas fuentes de datos diferentes de uno o más tipos diferentes.

Cada sistema de fuente/objetivo de datos y cada sistema de transmisión de datos en flujo 108 puede incluir su servicio perfilador (por ejemplo, servicio perfilador 112A-112E), servicio de monitorización (114A-114E), agente despachador (116A-116E) y/o servicio de transmisión en flujo (116A-116E) respectivo propio. En algunas formas de realización, cada fuente y objetivo de datos tiene la colección de servicios antes mencionada. En otras formas de realización, algunas fuentes de datos y algunos objetivos pueden tener los servicios, pero otros no los tendrán. Algunas fuentes de datos y algunos objetivos pueden tener solamente un conjunto parcial de los servicios. Todavía en otras formas de realización, ninguna fuente u objetivo utiliza dichos servicios.

El servicio perfilador 112A-112E captura estadísticas sobre lo que está ocurriendo en un sistema de ordenador (por ejemplo, memoria, CPU, utilización del disco, etcétera). Un servicio perfilador 112A-112E puede escribir datos en un catálogo de metadatos 110 o bien de manera directa o bien con la ayuda de procesos de metaservicio 140. En un ejemplo, un servicio perfilador 112A-112E puede recopilar datos de perfilado localmente además de escribir dichos datos en el catálogo de metadatos 110.

En una forma de realización, las fuentes de datos no se comunican directamente con destinos objetivo. En su lugar, el sistema de transmisión de datos en flujo 108 orquesta una transacción completa de transmisión en flujo en nombre de la fuente y el objetivo usando componentes y procesos compatibles con metadatos. En algunas formas de realización, los componentes y procesos, compatibles con metadatos, del sistema de transmisión de datos en flujo 108 no se comunican directamente con parte o con ninguno de los otros componentes o procesos del sistema. Por ejemplo, parte o la totalidad de dicha comunicación se puede producir de manera indirecta usando metadatos almacenados en el catálogo de metadatos 110.

El sistema de transmisión de datos en flujo 108 consulta al catálogo de metadatos 110 para determinar capacidades y disponibilidad de sistemas de ordenador implicados en/asociados a una transacción de transmisión de datos en flujo. Esto permite que el sistema de transmisión de datos en flujo 108 decida de manera inteligente cuándo iniciar

la transmisión de datos en flujo, por ejemplo, basándose en los recursos o poder computacionales disponibles, y cómo asignar eficazmente recursos cuando se lleva a cabo la transmisión de datos en flujo.

En uno de los ejemplos, un servicio perfilador 112A-112E almacena información de procesado del sistema de ordenador y/o de medios de almacenamiento de datos a intervalos regulares/diversos (por ejemplo, tiempo, eventos, etcétera), que se pueden basar en un ajuste de la configuración. Por ejemplo, un servicio perfilador 112A-112E puede obtener una instantánea de estadísticas de utilización en curso cada cinco segundos. Las estadísticas de utilización pueden incluir cualesquiera estadísticas asociadas a la utilización de la CPU, la utilización de la memoria, la utilización de los discos, la utilización de la red y/o la utilización de los medios de almacenamiento de datos. Dichas estadísticas se pueden almacenar, analizar, agregar y procesar adicionalmente a lo largo del tiempo para desarrollar estadísticas históricas, tales como líneas de base históricas.

En una forma de realización, se presenta un panel de control a los usuarios y/o administradores para proporcionar una instantánea del rendimiento en curso y/o estadísticas históricas. En un ejemplo, se proporciona una instantánea en curso en forma de una gráfica en línea con codificación de colores como parte de un panel de control *web*. Un operador/administrador puede hacer clic en un indicador que se vuelve rojo para ver diagnósticos asociados y para recibir información adicional sobre un tema. En otro ejemplo, se usa un panel de control *web* para presentar desviaciones (en tiempo real o previas) con respecto a los patrones de uso históricos.

Un servicio de monitorización 114A-114E es un agente que monitoriza otros servicios que se configuran para ejecutarse en un anfitrión particular. Por ejemplo, el servicio de monitorización 114A-114E puede determinar servicios que se supone que se están ejecutando en un sistema de ordenador particular (por ejemplo, fuentes/objetivos de datos tales como 102A, 102B, 102C, 104A, y el sistema de transmisión de datos en flujo 108, etcétera). En ejemplo, el servicio de monitorización 114A-114E hace *ping* en estos servicios de una manera periódica y, automáticamente, realiza una autoevaluación y reinicia todos los servicios que estén funcionando.

El agente despachador 116A-116E es responsable de recibir solicitudes de cliente. Las solicitudes pueden ser para ejecutar algún tipo de orden en un anfitrión particular que está haciendo funcionar un agente despachador 116A-116E. Por ejemplo, un nodo maestro o procesos de trabajo 160 del sistema de transmisión de datos en flujo 108 pueden llamar a un agente despachador respectivo 116A-116E para materializar una orden en el sistema en el que se ejecuta el agente despachador 116A-116E. En un ejemplo, se usa un agente despachador 116A-116E para llevar recuentos sobre una fuente/objetivo de datos y/o para determinar si u proceso de transmisión en flujo se ha completado de manera satisfactoria.

El servicio de transmisión en flujo 118A-118E es responsable de las operaciones “get” y “put”. En una forma de realización, el servicio de transmisión en flujo 118E en el sistema de transmisión de datos en flujo 108 se comunica con agentes de transmisión en flujo de cliente tales como el servicio de transmisión en flujo 118A, 118B, 118C y/o 118D. Los servicios de transmisión en flujo de cliente pueden enviar solicitudes para transmitir datos en flujo. Por ejemplo, un servicio de transmisión en flujo de cliente puede realizar una llamada al intermediario de solicitudes/respuestas 130 para transmitir datos en flujo entre sistemas.

En una forma de realización, los servicios de transmisión en flujo 118A-118E se pueden usar para recopilar datos de una fuente de datos local y para escribir datos en una fuente de datos de destino. Los servicios de transmisión en flujo 118A-118E pueden escribir el progreso y los resultados del trabajo que lleva a cabo cada servicio respectivo en el catálogo de metadatos 110. Además, los servicios de transmisión en flujo de cliente 118A-118D pueden trabajar en cooperación con el servicio de transmisión de flujo 118E del sistema de transmisión de datos en flujo 108 para transmitir datos en flujo.

El sistema de transmisión de datos en flujo 108 incluye servicios adaptadores de datos 120, el intermediario de solicitudes/respuestas 130, el proceso de metaservicio 140, el gestor de cargas de trabajo 150, los procesos de trabajo 160, los hilos de trabajo 170, el motor de generación de órdenes 180, el servicio perfilador 112E, el servicio de monitorización 114E, el agente despachador 116E y el servicio de transmisión en flujo 118E.

El sistema de transmisión de datos en flujo 108 orquesta y ejecuta servicios de transmisión de datos en flujo de alta velocidad, por ejemplo, entre plataformas de tecnología distribuida y no compatible. En un ejemplo, el sistema de transmisión de datos en flujo 108 usa uno o más nodos agrupados en *cluster* (por ejemplo, un *cluster* de Linux) para llevar a cabo operaciones. Por ejemplo, los nodos proporcionan los recursos informáticos usados para llevar a cabo diversas actividades que incluyen, aunque sin carácter limitativo, recibir y procesar solicitudes, analizar recursos, almacenar flujos de trabajo, determinar qué operaciones es necesario ejecutar, y transmitir datos en flujo.

En un entorno federado de transmisión de datos en flujo, cada sistema de transmisión de datos en flujo 108 entre una pluralidad de sistemas relacionados de transmisión de datos en flujo puede hacer que uno o más de sus propios nodos respectivos que son usados por él procesen datos. En un ejemplo, cada sistema de transmisión de datos en flujo 108 hace que al menos un nodo asociado lleve a cabo tareas. Los sistemas de transmisión de datos en flujo 108 con múltiples nodos pueden tener un único nodo maestro y múltiples nodos de trabajo. Además, un nodo maestro también puede actuar como nodo de trabajo en un entorno de nodo único o multinodo (por ejemplo,

un nodo maestro puede despachar trabajo para realizar a su propia dirección IP). En una forma de realización, los sistemas de transmisión de datos en flujo 108 no comparten ningún nodo. En otras formas de realización, los sistemas de transmisión de datos en flujo pueden compartir nodos y/o tomar prestados nodos (por ejemplo, tales como uno o más nodos de trabajo).

En arquitecturas de alta capacidad, alto rendimiento y/o alta disponibilidad, se pueden añadir nodos adicionales (para obtener un caudal, una velocidad, una tolerancia a fallos, etcétera, adicionales) usando una herramienta de interfaz gráfica de usuario (GUI) de etapa frontal o automáticamente de una reserva de recursos. Se pueden añadir nodos adicionales sin ningún tiempo de indisponibilidad usando una GUI o un proceso automatizado. En general, el número de nodos asociados al sistema de transmisión de datos en flujo 108 es dinámico ya que se pueden añadir o eliminar nodos de manera flexible mientras un sistema de transmisión de datos en flujo permanece en línea y operativo.

En uno de los ejemplos, el sistema de transmisión de datos en flujo 108 se proporciona en forma de un entorno federado. Por ejemplo, el sistema de transmisión de datos en flujo 108 puede existir como una pluralidad de sistemas diferentes de transmisión de datos en flujo interconectados que funcionan, cada uno de ellos, de manera independiente, pero comparten y transfieren trabajo de forma fluida. Por ejemplo, un primer sistema de transmisión de datos en flujo 108 se puede asignar en una primera región geográfica, y un segundo sistema de transmisión de datos en flujo 108 se puede asignar en una segunda región geográfica de entre una pluralidad de regiones geográficas a las que presta servicio un entorno federado.

En una forma de realización, el primer sistema de transmisión de datos en flujo 108 puede recibir una solicitud para transmitir datos en flujo entre dos medios de almacenamiento de datos en la segunda región geográfica. El primer sistema de transmisión de datos en flujo 108 puede tener conocimiento del segundo sistema de transmisión de datos en flujo 108 en o asociado a la segunda región geográfica, y puede transferir la solicitud al segundo sistema de transmisión de datos en flujo 108 para su procesado (por ejemplo, o bien antes o bien después de un proceso de autenticación o validación). De este modo, el primer sistema de transmisión de datos en flujo 108 puede transferir o asignar el trabajo al segundo sistema de transmisión de datos en flujo 108 en la configuración federada para garantizar que el trabajo se lleva a cabo de manera eficiente (por ejemplo, para no atravesar una red), segura, y/o para satisfacer requisitos específicos a nivel de servicio. Además, en un ejemplo, un sistema de transmisión de datos en flujo 108 puede orquestar una transmisión en flujo entre medios de almacenamiento de datos en un *cluster* local, en otro *cluster*, sobre centros de datos, o entre sistemas de ordenador (inclusive en el mismo sistema de ordenador).

Los servicios adaptadores de datos 120 son un conjunto de componentes de *software* que permiten que el sistema de transmisión de datos en flujo 108 se conecte e interaccione con varios tipos de fuentes de datos. Por ejemplo, puede haber disponibles uno o más servicios adaptadores de datos para integrar una fuente/objetivo de datos con el sistema de transmisión de datos en flujo 108. En uno de los ejemplos, se puede usar un adaptador de datos genérico, tal como un adaptador de Conectividad de Bases de Datos de Java (JDBC) para comunicarse con una fuente/objetivo de datos. En otro ejemplo, se puede desarrollar y utilizar un adaptador nativo desarrollado específicamente para una interacción y comunicación de alto rendimiento con una plataforma tecnológica asociada a una fuente de datos específica con el fin de proporcionar una transmisión de datos en flujo rápida/optimizada. El sistema de transmisión de datos en flujo 108 puede usar una arquitectura de sistema de transmisión de datos en flujo de alto rendimiento 100 que sea independiente de cualquier solución o plataforma tecnológica. Los servicios adaptadores de datos 120 permiten que el sistema de transmisión de datos en flujo 108 se interconecte con diversas soluciones técnicas no compatibles ofrecidas por diferentes proveedores.

El intermediario de solicitudes/respuestas 130 recibe y responde a solicitudes de clientes 101A, 101B. En un ejemplo, el intermediario de solicitudes/respuestas 130 es un componente escalable que, por ejemplo, permite que cientos o miles de clientes (por ejemplo, 101A y 101B) se conecten al sistema de transmisión de datos en flujo 108. Cuando el intermediario de solicitudes/respuestas 130 recibe una solicitud, el mismo puede analizar la solicitud para determinar la naturaleza de esta última (por ejemplo, transmisión en flujo, carga, extracción, duplicación, etcétera, de datos). A continuación, el intermediario de respuestas solicitudes 130 puede seleccionar una utilidad correspondiente y llamar a la misma para tratar la solicitud (por ejemplo, utilidad de transmisión en flujo, utilidad de carga, utilidad de extracción, utilidad de duplicación, etcétera, de datos).

El proceso de metaservicio 140 se comunica con un catálogo de metadatos 110. Por ejemplo, el proceso de metaservicio 140 puede leer datos del y escribir datos en el catálogo de metadatos 110. El catálogo de metadatos 110 puede contener información sobre seguridad, derechos, mapeo de datos, información de uso, recursos del sistema, etcétera, que están asociados al sistema de transmisión de datos de flujo 108. En un ejemplo, el catálogo de metadatos 110 también puede contener atributos de procesado físicos y/o lógicos y datos de rendimiento asociados a uno o más de las diversas fuentes/objetivos de datos (por ejemplo, 102A-C, 104A-C).

En un ejemplo, el proceso de metaservicio 140 consulta al catálogo de metadatos 110 para autenticar una solicitud de usuario basándose en título configurados de un usuario almacenados en el catálogo de metadatos 110. El proceso de metaservicio 140 también puede leer/escribir datos de actividad y rendimiento del sistema de

transmisión de datos en flujo 108 del/en el catálogo de metadatos 110.

El proceso de metaservicio 140 también puede almacenar información de mapeo de datos que permite que datos de una primera fuente/objetivo de datos (por ejemplo, 102A, 102B o 102C) se asocien o vinculen a una segunda fuente/objetivo de datos (por ejemplo, 104A, 104B o 104C). En un ejemplo, la información de mapeo permite la transferencia de datos entre (hacia y/o desde) la primera fuente/objetivo de datos y la segunda fuente/objetivo de datos. En un ejemplo, un mapeo o vinculación entre una fuente y un destino puede especificar uno o más sistemas específicos de transmisión de datos en flujo 108 (o federaciones/espacio de nombres) a usar para la transmisión en flujo de datos asociados al mapeo/vinculación.

El gestor de cargas de trabajo 150 es un proceso planificador interno asociado a un sistema de transmisión de datos en flujo 108. En un ejemplo, el gestor de cargas de trabajo 150 se ejecuta en un servidor maestro de un *cluster* de servidores que forman parte de un sistema de transmisión de datos en flujo 108. El gestor de cargas de trabajo 150 puede analizar metadatos de mapeo que describen asociaciones entre una primera fuente/objetivo de datos y una segunda fuente/objetivo de datos.

En un ejemplo, el gestor de cargas de trabajo 150 puede determinar que una tarea que recibe para transferir datos entre dos medios de almacenamiento de datos se debe ejecutar en al menos otro sistema de transmisión de datos en flujo 108 de entre una pluralidad de sistemas de transmisión de datos en flujo (por ejemplo, sobre la base de un acuerdo a nivel de servicio, una ubicación geográfica, una utilización esperada de recursos, recursos disponibles, una configuración de usuario/federación/espacio de nombres en un mapeo/vinculación, etcétera). El gestor de cargas de trabajo 150 también inicia o recurre a unos procesos de trabajo 160 para llevar a cabo un trabajo asociado a la solicitud recibida por el sistema de transmisión de datos en flujo 108.

En una forma de realización, cada sistema de transmisión de datos en flujo 108 de una pluralidad de sistemas de transmisión de datos en flujo tiene su propio gestor de cargas de trabajo 150 respectivo. Por ejemplo, cada sistema de transmisión de datos en flujo 108 puede tener su propio gestor de cargas de trabajo 150 que se ejecuta en el nodo maestro respectivo del sistema de transmisión de datos en flujo 108. En un ejemplo, cada gestor de cargas de trabajo 150 tiene su propio gobernador o límite en términos de un número total de tareas que puede llevar a cabo a la vez (independientemente de cuántas tareas se pudieran ejecutar físicamente en una fuente, objetivo y/o transmisión de datos en flujo en cualquier momento dado). Por ejemplo, un administrador puede configurar un parámetro de configuración del gestor de cargas de trabajo 150 del sistema de transmisión de datos en flujo 108 que defina un número máximo de tareas que puede ejecutar simultáneamente el gestor de cargas de trabajo particular.

En un ejemplo, el gestor de cargas de trabajo 150 actúa como un gobernador para todo el procesado asociado al sistema de transmisión de datos en flujo 108. Por ejemplo, incluso si los recursos externos pueden llevar a cabo treinta tareas en paralelo con respecto a una fuente, el gestor de cargas de trabajo 150 puede restringir el número máximo de tareas que puede ejecutar basándose en su propio máximo configurado (por ejemplo, menos de treinta), si fuera necesario. El gestor de cargas de trabajo 150 puede llevar a cabo una orquestación no solamente actuando como gobernador del procesado que se produce en el sistema de transmisión de datos en flujo 108, sino también gracias a que entiende (a través del catálogo de metadatos 110) lo que es capaz de realizar cada sistema y cuánto trabajo está llevando a cabo cada sistema en cualquier instante de tiempo dado. De este modo, el gestor de cargas de trabajo 150 puede despachar trabajo y ejecutar tareas de procesado de datos de manera inteligente basándose en dicha información.

El gestor de cargas de trabajo puede ser responsable de determinar cuándo deben llevarse a cabo operaciones solicitadas. En un ejemplo, el gestor de cargas de trabajo 150 realiza llamadas al catálogo de metadatos 110 para identificar solicitudes pendientes, estados de procesado, capacidades de procesado, cargas de trabajo existentes, etcétera, de sistemas de ordenador asociados a solicitudes pendientes cuando se despachan tareas. Por ejemplo, el gestor de cargas de trabajo 150 puede leer el catálogo de metadatos 110 para determinar estadísticas en curso sobre diversos aspectos de un sistema de origen, un sistema de destino objetivo, un sistema de transmisión de datos en flujo 108, una o más redes, equipos de red y/u otro u otros recursos informáticos. De este modo, el gestor de cargas de trabajo 150 puede llegar a tener conocimiento de la competencia (o disponibilidad) que existe en una máquina, en un equipo de red, en una o más redes, en un servidor de origen y/o en un servidor de destino.

En un ejemplo, el gestor de cargas de trabajo 150 identifica sistemas que se usarán para procesar una transacción de transmisión de datos en flujo sobre la base de un mapeo que se almacena en el catálogo de metadatos 110. El gestor de cargas de trabajo 150 se puede configurar también con parámetros para identificar cuántas tareas simultáneas se pueden ejecutar en cualquier momento dado en el sistema de transmisión de datos en flujo 108. Además, cada sistema de ordenador individual, tal como un sistema de origen u objetivo, puede tener sus propios parámetros asociados definidos en el catálogo de metadatos 110 que indican cuántas tareas simultáneas puede tratar un recurso respectivo o está configurado para tratar dicho recurso respectivo en un momento dado. El gestor de cargas de trabajo 150 puede considerar dicha información cuando se determina si despachar una solicitud pendiente de transmisión de datos en flujo para su procesado.

En un ejemplo, el gestor de cargas de trabajo 150 y/o el servicio perfilador 112A-112E puede identificar una situación crítica en uno o más sistemas asociados a un proceso pendiente o activo de transmisión en flujo. Por ejemplo, el gestor de cargas de trabajo 150 puede identificar proactivamente que un sistema se está quedando sin espacio libre o que un sistema está funcionando por encima de un umbral crítico de una CPU o una memoria. El gestor de cargas de trabajo 150 puede no despachar una solicitud pendiente de transmisión de datos en flujo para que se complete cuando existe dicha situación crítica. En su lugar, el gestor de cargas de trabajo 150 puede enviar una notificación de alerta a un administrador o usuario en relación con la condición y puede proporcionar una notificación de que la tarea no se procesará. En algunas formas de realización, el gestor de cargas de trabajo 150 interacciona con sistemas sobre centros de datos, ubicaciones geográficas y entidades comerciales diferentes.

En una forma de realización, los procesos de trabajo 160 reciben tareas que se despachan desde el gestor de cargas de trabajo 150 para llevar a cabo un trabajo asociado a una solicitud entrante de transmisión de datos en flujo. En un ejemplo, los procesos de trabajo 160 se pueden ejecutar en uno o más nodos de un *cluster* de nodos que están asociados al sistema de transmisión de datos en flujo 108.

Después de recibir una tarea que es despachada por el gestor de cargas de trabajo 150, un proceso de trabajo 160 puede actualizar el estado de la tarea despachada a activo. A continuación, un proceso de trabajo 160 puede analizar e inspeccionar los datos a transferir desde la primera fuente/objetivo de datos hasta una segunda fuente/objetivo de datos. Por ejemplo, un proceso de trabajo 160 puede analizar datos que se almacenan en el catálogo de metadatos 110 y que describen cómo está estructurada y segmentada en particiones física y/o lógicamente una carga útil de datos.

Un proceso de trabajo 160 puede analizar dichos metadatos para determinar cómo se pueden segmentar en particiones/dividir (y posteriormente ingresar) los datos en forma de una pluralidad de unidades de tamaño más pequeño, que pueden ser procesadas y transmitidas en flujo simultáneamente por diversos recursos informáticos que tienen diferentes capacidades. Además, un proceso de trabajo 160 puede invocar una pluralidad de hilos de trabajo 170 para llevar a cabo el procesado de los datos. El número de hilos de trabajo 170 invocados se puede basar en uno o más de los recursos disponibles del sistema en una fuente de datos, un destino objetivo, o un sistema de transmisión de datos en flujo 108.

En un ejemplo, el proceso de trabajo 160 es “compatible con particiones” lo cual significa que entiende cómo se almacenan físicamente los datos y puede determinar cómo se pueden segmentar en particiones lógica y/o físicamente los datos para facilitar un procesado paralelo. En un ejemplo, una partición física puede referirse a cómo están almacenados los datos en un archivo, en un sistema de archivos, o en una base de datos segmentada en particiones (por ejemplo, un archivo, diez archivos, cientos de archivos, miles de archivos, en una estructura de directorios particular basada en uno o más criterios, etcétera). En otro ejemplo, una partición lógica puede referirse a una forma de partir los datos basándose en un valor, tal como por fecha, productos, categorías, etcétera. Una partición física o lógica puede estar definida por un administrador como parte de la identificación de una estrategia de particiones para una fuente de datos particular. Dicha estrategia de particiones se puede almacenar en el catálogo de metadatos 110 para ayudar al proceso de trabajo 160 a determinar cómo procesar los datos.

En un ejemplo, el proceso de trabajo 160 determina un modelo de asignación o número de hilos a usar basándose en particiones físicas y/o lógicas identificadas para los datos. En un ejemplo, el proceso de trabajo 160 puede identificar particiones usando información que describe atributos y características de almacenamiento de los datos, que pueden estar disponibles en el catálogo de metadatos 110. En otro ejemplo, el proceso de trabajo 160 también puede detectar dinámicamente particiones o determinar cómo segmentar en particiones los datos analizando los propios datos, analizando metadatos que describen los datos, y/o analizando características de almacenamiento lógicas y físicas asociadas a los datos.

En un ejemplo, un proceso de trabajo 160 responsable de transferir datos desde 1000 archivos hasta un destino objetivo puede asignar o adjudicar cuatro hilos asíncronos para llevar a cabo la transferencia de datos. Por ejemplo, el proceso de trabajo 160 puede asignar 250 archivos a cada hilo para asignar uniformemente el trabajo entre los cuatro hilos. El proceso de trabajo 160 puede generar un manifiesto interno (archivo o metadatos) para ordenar qué archivos o grupo de archivos específicos debe procesar un hilo específico (por ejemplo, Hilo1 <1-250>, Hilo2 <251-500>, Hilo3 <501-750>, Hilo4 <751-1000>).

El proceso de trabajo 160 también puede asignar segmentos de datos a procesar en paralelo basándose en el tamaño, por ejemplo, cuando los segmentos de datos varíen de tamaño y no son uniformes. En un ejemplo, el proceso de trabajo 160 ordena archivos que se van a procesar según el tamaño y, a continuación, distribuye los archivos a cada hilo de una manera cíclica como método de distribución uniforme de la carga entre los hilos.

En una forma de realización, el gestor de cargas de trabajo 150 despacha una tarea a uno de una pluralidad de procesos de trabajo 160 basándose en una solicitud de transmisión de datos en flujo desde una fuente de datos hasta un destino objetivo. Por ejemplo, el gestor de cargas de trabajo 150 puede despachar una tarea a un proceso de trabajo 160 indirectamente actualizando el catálogo de metadatos 110 en lugar de hacerlo directamente llamando al proceso de trabajo 160. En un ejemplo, el gestor de cargas de trabajo 150 puede determinar que un

proceso de trabajo 160 está disponible leyendo el catálogo de metadatos 110 y puede asignar una tarea a ese proceso de trabajo 160 actualizando un campo de asignación de tareas asociado a un identificador exclusivo correspondiente (por ejemplo, un run_id) en el catálogo de metadatos 110.

Un proceso de trabajo 160 se puede ejecutar en uno o más nodos de trabajo en función de la configuración y/o de la capacidad disponible. En un ejemplo, un proceso de trabajo 160 modifica el estado de una tarea en el catálogo de metadatos 110 dependiente a activa cuando recibe la tarea. El proceso de trabajo 160 también puede analizar e inspeccionar los datos a transferir analizando información almacenada en el catálogo de metadatos 110 que describe cómo están estructurados y organizados los datos. Por ejemplo, un proceso de trabajo 160 puede determinar un factor de simultaneidad basándose en el análisis de cómo se pueden segmentar en particiones lógica y/o físicamente los datos de manera que dichas particiones se pueden procesar en paralelo cuando se transmiten los datos a un destino objetivo.

En un ejemplo, un proceso de trabajo 160 puede analizar atributos de almacenamiento de datos físicos, tales como una estructura de directorios, un número de archivos, y/o tamaños de archivos usados para almacenar datos cuando se determina una estrategia de particiones. Un proceso de trabajo 160 también puede analizar atributos lógicos de almacenamiento de datos, tales como tipos de tamaño o campo cuando se determina una estrategia de particiones. Además, el proceso de trabajo 160 puede analizar una muestra o un conjunto completo de datos para determinar cómo se estructuran, almacenan y/o distribuyen los datos cuando se determina una estrategia de particiones. En un ejemplo, el proceso de trabajo 160 determina cómo puede descomponerse un conjunto de datos en una pluralidad de fragmentos más pequeños que se pueden procesar eficientemente en paralelo entre una serie de diferentes recursos informáticos similares o no similares que tienen diversos niveles de disponibilidad y rendimiento.

En una forma de realización, el proceso de trabajo 160 determina un grado de paralelismo/factor de simultaneidad asociado a los datos. Por ejemplo, un proceso de trabajo 160 puede determinar que un conjunto de datos se puede dividir en cuatro, dieciséis, cientos o miles de fragmentos (por ejemplo, archivos, consultas, etcétera) para su procesado paralelo. A continuación, un proceso de trabajo 160 puede invocar uno o más hilos (es decir, hilos de trabajo 170) para llevar a cabo una transmisión en flujo paralela de los datos desde una fuente de datos hasta un destino objetivo. En un ejemplo, el proceso de trabajo 160 invoca una serie de hilos correspondientes a un determinado grado de paralelismo/factor de simultaneidad. De este modo, si el factor de simultaneidad es “cuatro”, el proceso de trabajo 160 puede invocar cuatro hilos para transmitir en flujo de los datos en paralelo.

En un ejemplo, el proceso de trabajo 160 puede analizar una carga útil para determinar cómo se pueden procesar en paralelo eficientemente datos asociados. En un ejemplo, el proceso de trabajo 160 despacha un único hilo de trabajo 170 cuando la carga útil es un único archivo/fracción de datos. Cuando hay muchos archivos/fracciones de datos, el proceso de trabajo 160 puede invocar una pluralidad de hilos basándose en un determinado grado de paralelismo/factor de simultaneidad, ajustes de configuración basados en el cliente, ajustes de configuración basados en el servidor y/o recursos informáticos disponibles de uno o más sistemas informáticos. A continuación, uno o más hilos de trabajo 170 asignados pueden transmitir en flujo la carga útil desde la fuente de datos hasta el destino objetivo. La transmisión en flujo puede incluir filtrar y/o transformar los datos a medida que se transfieren desde la fuente de datos hasta el destino objetivo. Los hilos de trabajo 170 asignados pueden llevar a cabo este trabajo basándose en metadatos de plan de ejecución almacenados en el catálogo de metadatos 110 que se generan dinámicamente obteniendo código ejecutable en tiempo de ejecución.

En un ejemplo, cada uno de entre una pluralidad de hilos de trabajo 170 invocados por un proceso de trabajo 160 para llevar a cabo una transmisión de datos en flujo paralela lee el catálogo de metadatos 110 para acceder a código de filtrado y/o transformación generado sobre la marcha en tiempo de ejecución a partir de metadatos de plan de ejecución almacenados en el catálogo de metadatos 110. El código de filtrado y/o transformación se genera basándose en un plan de ejecución creado por el gestor de cargas de trabajo 150 (por ejemplo, a partir de un mapeo de datos creado por el usuario, un flujo de trabajo, etcétera). En un ejemplo, cada hilo de trabajo 170 ensambla su propia versión (copia de trabajo o instanciación) de una secuencia de componentes operativos que usa para llevar a cabo diversas operaciones sobre los datos (por ejemplo, filtrado, agregación, transformación, depuración etcétera) a medida que los datos están siendo transmitidos en flujo.

En un ejemplo, se genera una secuencia de componentes operativos en tiempo de ejecución. Por ejemplo, un hilo de trabajo 170 puede ensamblar un conjunto de componentes operativos en una cadena donde la salida estándar de un componente se convierte en la entrada estándar del siguiente componente de la secuencia. De este modo, cada hilo puede procesar datos en paralelo como parte de una arquitectura de canalización (*pipeline*) (por ejemplo, cuando un primer fragmento de datos se ha trasladado desde una primera operación como salida estándar hasta una segunda operación ensamblada como entrada estándar, se procesa simultáneamente un segundo fragmento de datos usando la primera operación mientras se lleva a cabo la segunda operación sobre el primer fragmento de datos, y así sucesivamente). Pueden usarse múltiples capas de paralelismo para lograr un aumento sustancial del rendimiento, por ejemplo, cuando cada hilo de trabajo 170 procesa su propio conjunto de segmentos de datos de origen segmentados en particiones que se procesan también en paralelo con otros segmentos de datos de origen segmentados en particiones a lo largo de una canalización de componentes operativos conectados en cadena.

En una forma de realización, los hilos de trabajo 170 escriben periódicamente su progreso en el catálogo de metadatos 110. Un proceso de trabajo 160 que ha asignado los hilos de trabajo 170 también puede sondear periódicamente el catálogo de metadatos 110 para comprobar el estado de los hilos de trabajo 170. El proceso de trabajo 160 también puede analizar metadatos para determinar si cada uno de sus hilos de trabajo 170 ha completado su parte respectiva de la transmisión de datos en flujo de manera satisfactoria. En caso afirmativo, el proceso de trabajo 160 cambia el estado de la tarea completa en el catálogo de metadatos 110 de “activa” a “final”. Por otro lado, y en función de la situación, el hilo de trabajo puede actualizar el estado a “fallido” si cualquiera de los hilos de trabajo 170 asociados no se completase satisfactoriamente.

En un ejemplo, el proceso de trabajo 160 puede detener tareas de larga ejecución o bien automáticamente o bien basándose en una solicitud de usuario y puede actualizar el estado de la tarea a “abortada” o “cancelada”. El proceso de trabajo 160 también puede actualizar el estado a “vacía” cuando no se producen datos desde la fuente, por ejemplo, porque no existen datos en la fuente o porque no se produjeron datos resultantes cuando se aplicó un filtro.

En una forma de realización, se comprimen datos de origen en un sistema de origen para reducir el tamaño de datos a transferir a través de una red, por ejemplo, cuando se transmiten en flujo directamente datos de origen al servidor de destino sin ninguna manipulación. A continuación, se pueden descomprimir los datos en el destino de forma correspondiente, si ello fuese necesario. En otro ejemplo, se descomprimen en la fuente datos comprimidos en la fuente cuando el gestor de cargas de trabajo 150 determina que debe producirse un filtrado y/o cualquier tipo de transformación antes de que los datos lleguen al destino objetivo.

El motor de generación de órdenes 180 genera funciones que se materializan en tiempo de ejecución sobre la base de metadatos. Las funciones creadas por el motor de generación de órdenes 180 son ensambladas y ejecutadas por cada hilo de trabajo 170, permitiendo así que cada hilo funcione eficazmente como un motor de transformación compartimentado con acceso a una biblioteca de su propio conjunto de funciones ligeras para rendimiento optimizado.

En una forma de realización, el motor de generación de órdenes 180 lee el catálogo de metadatos 110 para determinar funciones a generar para hilos de trabajo 170 que se han invocado o se invocarán con el fin de llevar a cabo un plan de ejecución generado por el gestor de cargas de trabajo 150. En un ejemplo, el gestor de cargas de trabajo 150 puede determinar que se lleve a cabo un conjunto de operaciones sobre un conjunto de datos que se van a transmitir en flujo desde una fuente de datos hasta un destino objetivo. Las operaciones se pueden definir, por ejemplo, como parte de un mapeo de datos, o de manera adicional a este último, entre uno o más campos de una ubicación de almacenamiento de datos de una fuente de datos y uno o más campos de una ubicación de almacenamiento de datos de un destino objetivo. El gestor de cargas de trabajo 150 puede producir un plan de ejecución para llevar a cabo la transmisión en flujo, comprendiendo el plan de ejecución un conjunto de funciones (por ejemplo, transformación, filtrado, personalización, etcétera) a ejecutar en secuencia por cada uno de uno o más hilos de trabajo 170.

En un ejemplo, un plan de ejecución generado por el gestor de cargas de trabajo 150 se puede representar como un conjunto de parámetros de configuración o en un formato XML que puede ser procesado y ejecutado por una o más versiones diferentes de un motor del sistema de transmisión de datos en flujo 108. Por ejemplo, el sistema de transmisión de datos en flujo 108 puede generar, ejecutar y/o poner en funcionamiento instrucciones u órdenes del plan de ejecución que se representan como un conjunto de parámetros personalizados o en un formato XML personalizado.

La figura 2 es un diagrama de flujo que ilustra una transmisión de datos en flujo de alto rendimiento, según una forma de realización. El método 200 puede ser llevado a cabo por un módulo lógico de procesado que puede comprender *hardware* (circuitería, módulo lógico dedicado, módulo lógico programable, microcódigo, etcétera), *software* (tal como instrucciones ejecutadas en un sistema de ordenador de propósito general, una máquina dedicada o un dispositivo de procesado), microprogramas o una combinación de los mismos. En un ejemplo, el método 200 se lleva a cabo usando el sistema de transmisión de datos en flujo 108 de la figura 1.

En la etapa 210, se recibe un mapeo de datos que describe una asociación entre una fuente de datos y un destino objetivo. En una forma de realización, uno o más elementos de datos de una ubicación de almacenamiento de una primera fuente de datos se asocian o vinculan a uno o más elementos de datos de una ubicación de almacenamiento de un destino objetivo (por ejemplo, elementos de archivos de datos, campos de bases de datos, datos XML, campos de datos en formatos de datos personalizados, etcétera). En general, se puede recibir cualquier mapeo de datos que describa una asociación o relación entre dos o más elementos de datos, campos, contenedores, archivos u otras estructuras de datos.

En un ejemplo, se pueden mapear directamente datos de una fuente de datos a un destino objetivo. También se pueden definir transformaciones de datos para modificar datos de una fuente de datos a medida que se están transmitiendo en flujo datos a un destino objetivo. Por ejemplo, se pueden definir una o más transformaciones de

datos como parte de un mapeo de datos. Las transformaciones de datos se pueden configurar para modificar datos de origen, por ejemplo, combinando una pluralidad de campos de datos de origen en un campo de destino objetivo, dividiendo campos de datos de origen en múltiples campos de destino objetivo, filtrando datos de origen, depurando datos de origen, etcétera. Dicho mapeo y transformaciones se pueden proporcionar en un flujo de trabajo definido por el usuario, configurado para transformar datos de una fuente de datos cuando se transmiten en flujo los datos a un destino objetivo.

En la etapa 220, se genera un plan de ejecución de transferencia de datos a partir del mapeo de datos para transferir los datos desde la fuente de datos hasta el destino objetivo. En una forma de realización, el motor de flujo de trabajo 150 analiza un mapeo de datos que describe una asociación entre una fuente de datos y un destino de almacenamiento objetivo. El mapeo de datos puede incluir transformaciones de datos y otras operaciones que se deben realizar cuando se transmiten datos en flujo desde la fuente de datos hasta el destino objetivo.

En una forma de realización, el motor de flujos de trabajo 150 genera y almacena un plan de ejecución para mapeos y transformaciones de datos en forma de metadatos en el catálogo de metadatos 110. En un ejemplo, los metadatos del plan de ejecución generados por el motor de flujos de trabajo 150 se pueden procesar, interpretar y/o ejecutar por medio de una o más versiones diferentes del sistema de transmisión de datos en flujo 108. Los metadatos también se pueden usar para generar un código ejecutable, que puede ser ejecutado por cualquier proceso (por ejemplo, hilos de trabajo 170).

En la etapa 230, se transfieren datos desde la fuente de datos hasta el destino objetivo usando el plan generado de ejecución de transferencia de datos. En una forma de realización, el sistema de transmisión de datos en flujo 108 usa un plan de ejecución de transferencia de datos almacenado en el catálogo de metadatos 110 para generar código ejecutable en tiempo de ejecución. A continuación, el sistema de transmisión de datos en flujo puede ejecutar el código ejecutable generado a partir de los metadatos del plan de ejecución de transferencia de datos usando hilos de trabajo 170. De este modo, los hilos de trabajo 170 pueden ejecutar el código generado en tiempo de ejecución para transmitir en flujo datos desde una fuente de datos hasta un destino objetivo.

La figura 3 es un diagrama de flujo que ilustra otros aspectos de la transmisión de datos en flujo de alto rendimiento, según una forma de realización. El método 300 puede ser llevado a cabo por un módulo lógico de procesado que puede comprender *hardware* (circuitería, módulo lógico dedicado, módulo lógico programable, microcódigo, etcétera), *software* (tal como instrucciones ejecutadas en un sistema de ordenador de propósito general, una máquina dedicada o un dispositivo de procesado), microprogramas o una combinación de los mismos. En un ejemplo, el método 300 se lleva a cabo usando el sistema de transmisión de datos en flujo 108 de la figura 1.

En la etapa 310, se recibe información sobre una primera fuente de datos. En la etapa 320, se recibe información sobre un destino objetivo. En una forma de realización, un usuario registra una fuente de datos en un cliente 101A, 101B, usando una interfaz gráfica de usuario (GUI) como parte de un proceso de registro. El sistema de transmisión de datos en flujo 108 puede descubrir registrar automáticamente también una o más fuentes de datos.

Como parte de un proceso de descubrimiento o registro, se puede asimilar o adquirir información sobre una fuente de datos. Por ejemplo, la información de la fuente de datos puede incluir uno o más de recursos informáticos fijos o disponibles de un sistema de ordenador que aloja la fuente de datos, información sobre el tipo de fuente de datos (por ejemplo, base de datos relacional, base de datos en memoria, base de datos relacional de objetos, sistema de archivos, aparato, etcétera), información de proveedores, información de la versión, ajustes de configuración del sistema de ordenador y/o de la fuente de datos, disponibilidad del sistema de ordenador y/o características o compatibilidad de la fuente de datos, etcétera.

En un ejemplo, una primera fuente de datos puede incluir uno o más de fuentes/objetivos de datos 102A-102C, y una segunda fuente de datos puede incluir uno o más de los recursos/objetivos 104A-104C (o viceversa). Además, la información recibida, recogida, descubierta o adquirida sobre una primera fuente de datos y/o una segunda fuente de datos se puede almacenar en forma de metadatos en el catálogo de metadatos 110 para una posterior referencia por parte del sistema de transmisión de datos en flujo 108.

En la etapa 330, se recibe un mapeo de datos que asocia una ubicación de almacenamiento de datos de la primera fuente de datos a una ubicación de almacenamiento de datos del destino objetivo. En un ejemplo, los elementos de datos identificados de una primera fuente de datos se asocian o vinculan con elementos de datos identificados que existen en una segunda fuente/destino objetivo de datos (por ejemplo, campos de una base de datos). En general, se puede recibir un mapeo, asociación o relación entre dos elementos de almacenamiento de datos, campos, contenedores, archivos, etcétera, cualesquiera.

En un ejemplo, un usuario puede designar un mapeo entre una fuente de datos y un destino objetivo como un flujo de trabajo que transforma datos de un formato del sistema de origen en una forma que es compatible con un sistema de destino objetivo (por ejemplo, usando un diseñador de flujos de trabajo). De este modo, un mapeo puede comprender operaciones que se usan para modificar datos que se van a transferir o copiar desde un sistema de origen al sistema de destino objetivo (por ejemplo, como parte de una tarea de transmisión de datos en flujo).

En otra forma de realización, se pueden mapear datos de origen basándose en un patrón detectado. Por ejemplo, si un administrador o preproceso no ha creado un mapeo de metadatos, dicho mapeo se puede generar en tiempo real. El mapeo de datos en tiempo real se puede basar en una o más áreas temáticas (por ejemplo, transacciones) que se identifiquen como parte de una solicitud. Las áreas temáticas se pueden usar para buscar en metadatos dinámicamente en tiempo real, por ejemplo, usando datos de origen identificados como coincidentes con una o más áreas temáticas particulares.

En un ejemplo, un proceso de trabajo 160 puede ejecutar una búsqueda por patrones basada en un área temática y llevar a cabo una inspección sobre datos asociados a un área temática particular para determinar cuántos hilos de trabajo 170 invocará para procesar los datos identificados dinámicamente. Por ejemplo, un proceso de trabajo 160 puede tener conocimiento de que los datos de transacciones para cada uno de una pluralidad de instrumentos financieros están almacenados en un archivo respectivo correspondiente a cada uno de los instrumentos. El proceso de trabajo 160 puede descubrir y/o se le puede ordenar que procese un conjunto total o parcial de los datos de transacciones. En un ejemplo, el proceso de trabajo 160 asigna un hilo de trabajo 170 para transmitir en flujo cada respectivo archivo diferente que se le ha ordenado procesar.

En la etapa 340, se recibe una solicitud para transferir datos desde la primera fuente de datos hasta el destino objetivo basándose en el mapeo. En una forma de realización, se recibe una solicitud para mover datos entre un sistema de origen y un sistema objetivo. Por ejemplo, la solicitud para transmitir en flujo, cargar, extraer y/o duplicar datos entre uno o más sistemas de ordenador se puede recibir en una solicitud. Por ejemplo, el intermediario de solicitudes/respuestas 130 puede recibir una solicitud para mover datos entre dos centros de datos diferentes. En un ejemplo, una solicitud puede identificar recursos de datos lógicos y/o físicos que se transmitirán en flujo (por ejemplo, transferirán, copiarán, etcétera) de una fuente de datos a un destino objetivo.

En una forma de realización, el intermediario de solicitudes/respuestas 130 analiza una solicitud entrante para determinar información sobre la solicitud. Por ejemplo, el intermediario de solicitudes/respuestas 130 puede determinar el tipo de solicitud que se recibe de manera que puede llamar a una utilidad, componente o servicio de procesado correspondiente adecuado. En un ejemplo, el intermediario de solicitudes/respuestas 130 puede invocar una utilidad de transmisión en flujo que valida y autentica la solicitud. Por ejemplo, la utilidad de transmisión en flujo puede autenticar a un usuario particular que inicia la solicitud y confirmar que la solicitud es válida.

En un ejemplo, una solicitud de transmisión en flujo remite a un mapeo predefinido entre dos medios de almacenamiento de datos. Una utilidad de transmisión en flujo lleva a cabo una búsqueda en el catálogo de metadatos 110 para determinar si dicho mapeo existe. En caso afirmativo, la utilidad de transmisión en flujo usa el catálogo de metadatos 110 para validar adicionalmente la solicitud. Por ejemplo, una utilidad de transmisión en flujo puede validar una dimensionalidad en el tiempo de un conjunto de datos solicitado antes de intentar realmente transmitir en flujo el conjunto de datos.

En una forma de realización, una vez que se ha autenticado una solicitud de usuario y se ha validado un mapeo de datos considerando los datos solicitados, la solicitud se sitúa en cola de espera para su procesar y su estado se actualiza a “pendiente”. En un ejemplo, se genera un identificador exclusivo run_id para una solicitud que ha sido presentada para procesarse. Por ejemplo, un proceso de metaservicio 140 puede generar un identificador exclusivo de 24 dígitos para la solicitud. A continuación, el proceso de metaservicio 140 puede introducir la solicitud en el catálogo de metadatos 110 para permitir el seguimiento de la solicitud y sus datos asociados durante todo el ciclo de vida de la solicitud.

Una vez que se ha enviado una solicitud para un procesado en el catálogo de metadatos 110, procesos de metaservicio 140 pueden registrar en el catálogo de metadatos 110 información asociada a la solicitud. Por ejemplo, en catálogo de metadatos 110 se pueden almacenar una ID de usuario, una cuenta de usuario, una ID de aplicación, una dirección IP en la que se originó la solicitud, un tipo de solicitud, información de vinculación (relación/mapeo de fuente-a-objetivo) y otra información y detalles sobre la solicitud. Además, el estado de la solicitud se puede actualizar a un estado “pendiente”, que identificará la solicitud como disponible para su procesado en una lista de solicitudes situadas en cola de espera que son analizadas por el gestor de cargas de trabajo 150.

En un ejemplo, el gestor de cargas de trabajo 150 busca solicitudes en un estado “pendiente” que estén preparadas para el procesado. Además, cuando el gestor de cargas de trabajo 150 tiene disponibles procesos de trabajo 160, puede asignar un proceso de trabajo 160 para completar una solicitud “pendiente”.

En la etapa 350, se genera un plan de ejecución de transferencia de datos basándose en un mapeo de datos. En un ejemplo, el motor de flujos de trabajo 150 analiza información de mapeo que describe una asociación entre una fuente de datos y un destino de almacenamiento objetivo. El mapeo puede incluir, se puede basar o puede usar para generar un flujo de trabajo o secuencia de pasos interconectados que se pueden utilizar para procesar datos de origen de manera que sean compatibles y encajen dentro del paradigma del destino de almacenamiento objetivo (lógica y/o físicamente). Por ejemplo, como parte del proceso de transmisión en flujo puede que sea necesario

filtrar, analizar sintácticamente, transformar, convertir, etcétera, los datos de origen.

En un ejemplo, el gestor de cargas de trabajo 150 genera y almacena un plan de ejecución que permite que uno o más hilos de trabajo 170 construyan o ensamblen una serie de órdenes usadas para ejecutar el proceso (mapeo/flujo de trabajo) en tiempo de ejecución. En un ejemplo, el gestor de cargas de trabajo 150 genera un plan de ejecución en forma de un conjunto de datos con formato XML, que se almacenan en el catálogo de metadatos 110.

En la etapa 360, los datos se transfieren desde la primera fuente de datos hasta el destino objetivo en paralelo sobre la base del plan de ejecución de transferencia de datos. En una forma de realización, en el catálogo de metadatos 110 se almacena un plan de ejecución de transferencia de datos generado por el gestor de cargas de trabajo. El plan de ejecución de transferencia de datos puede incluir información que permite que hilos de trabajo 170 lleven a cabo operaciones de flujo de trabajo/mapeo de datos a medida que se transmiten datos en flujo desde una fuente hasta un destino objetivo.

Por ejemplo, como parte de un proceso de transmisión en flujo puede que sea necesario filtrar datos recuperados de una fuente de datos con petición previa. Adicionalmente, puede que sea necesario transformar los datos de una o más maneras para permitir que los mismos sean compatibles en cuanto a forma (por ejemplo, físicamente) o en cuanto a sustancia (por ejemplo, lógicamente) sobre la base de una configuración de destino objetivo. De este modo, puede que sea necesario modificar los datos de diversas maneras, que pueden incluir, aunque sin carácter limitativo, concatenación, truncamiento, sustitución, actualizaciones, funciones personalizadas, etcétera.

Dependiendo de cómo se diseñe un mapeo o flujo de trabajo particular, puede que sea necesario llevar a cabo estas operaciones en una secuencia particular. Además, se pueden utilizar operaciones convencionales o personalizadas (por ejemplo, funciones, procedimientos, etcétera, definidos por el usuario). En una forma de realización, un usuario puede crear funciones y procedimientos personalizados y los mismos se pueden integrar en el flujo de trabajo/mapeo de datos en forma de uno o más pasos ordenados. En un ejemplo, un usuario puede definir funciones/procedimientos personalizados en un lenguaje de secuencia de instrucciones (*scripting*) privativo (por ejemplo, secuencia de instrucciones de transmisión de datos en flujo bajo demanda). En otro ejemplo, un usuario también puede definir funciones/procedimientos personalizados usando el lenguaje de consulta estructurado (SQL) u otro lenguaje de ordenador.

En una forma de realización, después de que el gestor de cargas de trabajo 150 genere un plan de ejecución y encuentre un proceso de trabajo 160 disponible para tratar una solicitud entrante, el gestor de cargas de trabajo 150 asigna la solicitud al proceso de trabajo 160 disponible.

En un ejemplo, un proceso de trabajo 160 determina cómo procesará trabajo asociado a la solicitud. Por ejemplo, el proceso de trabajo 160 puede analizar la carga útil de los datos de origen que es necesario procesar. El proceso de trabajo 160 puede analizar datos de origen para determinar cómo los datos se pueden segmentar en particiones o podar de manera física, lógica, horizontal, vertical, etcétera. El proceso de trabajo 160 puede analizar datos de origen basándose en información del catálogo de metadatos 110, o accediendo a los datos directamente (por ejemplo, por muestreo, examinando cómo están almacenados los datos, etcétera). El proceso de trabajo 160 también puede analizar la utilización y la capacidad de la máquina de origen, así como la utilización y la capacidad de la máquina objetivo. El proceso de trabajo 160 puede usar esta información para determinar un grado de paralelismo que se puede usar para procesar datos de origen en paralelo.

En una forma de realización, el proceso de trabajo 160 invoca uno o más hilos de trabajo 170 asíncronos, con los cuales no está en comunicación directamente. En un ejemplo, el proceso de trabajo 160 puede interaccionar con hilos asociados indirectamente leyendo y/o escribiendo metadatos almacenados en el catálogo de metadatos 110. Los hilos de trabajo 170, por ejemplo, se pueden ejecutar en el mismo nodo o pueden estar diseminados entre diferentes nodos de trabajo asociados a uno o más sistemas de transmisión de datos en flujo 108 en un entorno federado. Los hilos de trabajo 170 pueden llevar a cabo las operaciones que es necesario realizar para completar una solicitud asignada por el gestor de cargas de trabajo 150. En un ejemplo, los hilos de trabajo 170 ejecutan operaciones definidas en un plan de ejecución creado por el gestor de cargas de trabajo 150.

En una forma de realización, el catálogo de metadatos 110 almacena un listado de cada procedimiento/función (incluida secuencia de procesado) asociado a un plan de ejecución para una tarea que ha sido generada por el gestor de cargas de trabajo 150. En un ejemplo, el motor de generación de órdenes 180 genera funciones que se materializan en tiempo de ejecución basándose en metadatos almacenados en el catálogo de metadatos por el gestor de cargas de trabajo. Las funciones creadas por el motor de generación de órdenes 180 son ensambladas y ejecutadas por hilos de trabajo 170, con lo cual se permite que cada hilo funcione efectivamente como un motor de transformación autónomo con acceso a una biblioteca de su propio conjunto respectivo de funciones ligeras.

En una forma de realización, el motor de generación de órdenes 180 analiza un plan de ejecución de transferencia de datos generado por el gestor de cargas de trabajo 150 almacenado en el catálogo de metadatos 110. A continuación, el motor de generación de órdenes 180 genera fragmentos de código (funciones/procedimientos)

que son ejecutables por hilos de trabajo 170.

En una forma de realización, el motor de generación de órdenes 180 puede generar código ejecutable para funciones/procedimientos convencionales proporcionados por el sistema. El motor de generación de órdenes 180 también puede generar código ejecutable para funciones y procedimientos definidos por el usuario escritos en el lenguaje de ordenador tal como un lenguaje de secuencias de instrucciones privativo o el lenguaje de consulta estructurada (SQL). En un ejemplo, el motor de generación de órdenes 180 genera código en tiempo de ejecución y al mismo le pueden llamar procesos de trabajo 160. En un ejemplo, el motor de generación de órdenes 180 puede generar código en cualquier momento.

En una forma de realización, cada hilo de trabajo 170 asignado a una tarea usa código generado por el motor de generación de órdenes 180 para configurar una instancia autónoma, respectiva, de un motor de mapeo/transformación/flujo de trabajo con el fin de procesar y transmitir en flujo datos de origen a un destino objetivo. En un ejemplo, los hilos de trabajo 170 ensamblan los fragmentos ejecutables de código generados por el motor de generación de órdenes 180 de acuerdo con un plan de ejecución creado previamente por el gestor de cargas de trabajo 150.

En una forma de realización, los hilos de trabajo 170 ensamblan fragmentos ejecutables de código en una secuencia y de una manera definidas por un plan de ejecución. Cada hilo de trabajo puede ensamblar fragmentos de código ejecutables diferentes encadenando entre sí los diferentes fragmentos de código. Por ejemplo, el primer fragmento de código (componente) ejecutable puede recibir la unidad de datos como entrada convencional. A continuación, los hilos de trabajo 170 pueden encadenar el primer componente con un segundo componente de manera que la salida convencional del primer componente alimenta la entrada convencional del segundo componente. Continuando con este ejemplo no limitativo, a continuación, la salida convencional del segundo componente puede alimentar la entrada convencional del tercer componente, y así sucesivamente. De esta manera, la entrada de cualquier función es la salida de la función previa durante toda una secuencia completa.

En el ejemplo previo, cada hilo de trabajo se convierte efectivamente en un motor de transformación con acceso a una biblioteca de funciones que se materializan en tiempo de ejecución. De este modo, los datos de origen se pueden procesar sin fisuras a medida que son transmitidos en flujo en un entorno sin estado y sin ningún bloqueo.

En la etapa 370, se proporciona un manifiesto que comprende información que describe la transferencia de datos. En una forma de realización, al completarse satisfactoriamente una tarea de transmisión de datos en flujo se escribe un manifiesto de entrega. En un ejemplo, se escriben manifiestos de entrega idénticos o diferentes en un sistema de origen y en un sistema de destino objetivo cuando la tarea de transmisión de datos en flujo ha finalizado satisfactoriamente. El manifiesto de entrega puede incluir uno o más de entre una descripción de los datos que se entregaron, un tiempo de inicio, un tiempo final, archivos que se entregaron, un tamaño de cada archivo entregado, características de cada archivo entregado, etcétera.

En un ejemplo, un manifiesto de entrega incluye también un estado e información sobre el formateo de los datos que se entregaron. Por ejemplo, dicha información puede incluir un delimitador, mensajes de error, formateo de datos (por ejemplo, tipos de campo, formatos de tiempo, formatos de datos, formatos numéricos, valores NULL, uso de caracteres especiales), etcétera.

En un ejemplo, un planificador de tareas en un destino objetivo buscará un archivo de manifiesto de entrega antes de comenzar a procesar cualesquiera datos entrantes. En algunas formas de realización, esto garantiza que el planificador de tareas no dé inicio a un procesado subsiguiente de forma prematura porque el manifiesto de entrega puede ser el último fragmento de información que se escribe cuando se ha completado una tarea de transmisión en flujo.

En una forma de realización, en el manifiesto de entrega se describen datos que se han entregado al destino objetivo, lo cual permite transferir los datos a una organización de aguas abajo sin requerir ningún cambio sobre el sistema de transmisión de datos en flujo 108. En un ejemplo, se describen datos que han sido entregados y los mismos pueden ser procesados por una organización receptora basándose en la descripción proporcionada por el manifiesto. De este modo, los cambios de aguas arriba sobre datos de origen por parte de una organización no deberían tener impacto alguno sobre las operaciones en el sistema de transmisión de datos en flujo 108 ya que la organización que recibe los datos en el destino objetivo puede fiarse del manifiesto de destino generado para una tarea de transmisión de datos en flujo.

La figura 4 ilustra un diagrama de una máquina en la forma ejemplificativa de un sistema de ordenador 400 dentro del cual se puede ejecutar un conjunto de instrucciones, para provocar que la máquina lleve a cabo una cualquiera o más de las metodologías descritas en la presente. En formas de realización alternativas, la máquina se puede conectar (por ejemplo, en red) a otras máquinas en una LAN, una intranet, una extranet, o Internet. La máquina puede funcionar en calidad de servidor o máquina de cliente en un entorno de red cliente-servidor, o como una máquina par en un entorno de red entre entidades pares (o distribuido). La máquina puede ser un ordenador personal (PC), un PC de tipo tableta, una caja de adaptación del televisor (STB), un Asistente Personal Digital

(PDA), un teléfono celular, un aparato *web*, un servidor, un rúter de red, un conmutador o puente, o cualquier máquina capaz de ejecutar un conjunto de instrucciones (secuencial o de otro tipo) que especifiquen acciones a realizar por esa máquina. Además, aunque se ilustra solamente una única máquina, se considerará también que el término "máquina" incluye cualquier conjunto de máquinas que, de manera individual o conjunta, ejecuten un conjunto (o múltiples conjuntos) de instrucciones para llevar a cabo una o más cualesquiera de las metodologías descritas en la presente memoria.

El sistema de ordenador 400 ejejmplficaitvo incluye un dispositivo de procesado (procesador) 402, una memoria principal 404 (por ejemplo, memoria de solo lectura (ROM), memoria *flash*, memoria dinámica de acceso aleatorio (DRAM) tal como una DRAM síncrona (SDRAM), una SDRAM de doble velocidad de datos (DDR) o una DRAM (RDRAM), etcétera), una memoria estática 406 (por ejemplo, memoria *flash*, memoria estática de acceso aleatorio (SRAM), etcétera), y un dispositivo de almacenamiento de datos 418, que se comunican entre sí por medio de un bus 430.

El procesador 402 representa uno o más dispositivos de procesado de propósito general tales como un microprocesador, una unidad de procesado central o similares. Más particularmente, el procesador 402 puede ser un microprocesador de computación con conjunto complejo de instrucciones (CISC), un microprocesador de computación con conjunto reducido de instrucciones (RISC), un microprocesador de palabras de instrucciones muy largas (VLIW), o un procesador que implemente otros conjuntos de instrucciones o procesadores que implementen una combinación de conjuntos de instrucciones. El procesador 402 también puede ser uno o más dispositivos de procesado de propósito especial, tales como un circuito integrado de aplicación específica (ASIC), una matriz de puertas programable in situ (FPGA), un procesador de señal digital (DSP), un procesador de red, o similares. El procesador 402 está configurado para ejecutar instrucciones 422 con el fin de llevar a cabo las operaciones y pasos descritos en la presente.

El sistema de ordenador 400 puede incluir, además, un dispositivo de interfaz de red 408. El sistema de ordenador 400 también puede incluir una unidad de visualización de vídeo 410 (por ejemplo, una pantalla de cristal líquido (LCD) o un tubo de rayos catódicos (CRT)), un dispositivo de entrada alfanumérica 412 (por ejemplo, un teclado), un dispositivo de control por cursor 414 (por ejemplo, un ratón) y un dispositivo de generación de señales 416 (por ejemplo, un altavoz).

El dispositivo de almacenamiento de datos 418 puede incluir un soporte de almacenamiento legible por ordenador 428 en el cual se almacenan uno o más conjuntos de instrucciones 422 (por ejemplo, *software*) que materializan una o más cualesquiera de las metodologías o funciones descritas en la presente. Las instrucciones 422 también pueden residir, de manera completa o al menos parcialmente, dentro de la memoria principal 404 y/o dentro del procesador 402 durante la ejecución de las mismas por parte del sistema de ordenador 400, constituyendo también la memoria principal 404 y el procesador 402 soportes de almacenamiento legibles por ordenador. Las instrucciones 422 además se pueden transmitir o recibir a través de una red 420 por medio del dispositivo de interfaz de red 408.

En una forma de realización, las instrucciones 422 incluyen instrucciones para una arquitectura de sistema de transmisión de datos en flujo de alto rendimiento 100 (por ejemplo, el sistema de transmisión de datos en flujo 108 de la figura 1) y/o una biblioteca de *software* que contiene métodos que llaman a un sistema de transmisión de datos en flujo 108. Aunque, en una forma de realización ejemplificativa, el soporte de almacenamiento legible por ordenador 428 (soporte de almacenamiento legible por máquina) se muestra de manera que es un único soporte, debe considerarse que el término "soporte de almacenamiento legible por ordenador" incluye un único soporte o múltiples soportes (por ejemplo, una base de datos centralizada o distribuida, y/o memorias caché y servidores asociados) que almacenan el conjunto o conjuntos de instrucciones. También debe considerarse que el término "soporte de almacenamiento legible por ordenador" incluye cualquier soporte que sea capaz de almacenar, codificar o llevar un conjunto de instrucciones para su ejecución por parte de la máquina y que provocan que la máquina lleve a cabo una o más cualesquiera de las metodologías de la presente invención. Por consiguiente, se considerará que el término "soporte de almacenamiento legible por ordenador" incluye, aunque sin carácter limitativo, memorias de estado sólido, soportes ópticos y soportes magnéticos.

En la descripción anterior, se exponen numerosos detalles. No obstante, se pondrá de manifiesto para alguien con conocimientos en la materia que pueda sacar provecho de esta exposición, que la presente invención puede ponerse en práctica sin estos detalles específicos. En algunos casos, estructuras y dispositivos ampliamente conocidos se muestran en forma de diagrama de bloques, más que de manera detallada, con el fin de evitar entorpecer la presente invención.

Algunas partes de la descripción detallada se han presentado en términos de algoritmos y representaciones simbólicas de operaciones sobre bits de datos dentro de una memoria de ordenador. En la presente memoria, y de manera general, se concibe un algoritmo de manera que es una secuencia autónoma de pasos que conducen a un resultado deseado. Los pasos son aquellos que requieren manipulaciones físicas de magnitudes físicas. Habitualmente, aunque no de forma necesaria, estas magnitudes adoptan la forma de señales eléctricas o magnéticas con capacidad de ser almacenadas, transferidas, combinadas, comparadas y manipuladas de otro

modo. En ocasiones se ha demostrado que es conveniente, por motivos de uso común, referirse a estas señales como bits, valores, elementos, símbolos, caracteres, términos, números o similares.

No obstante, debe tenerse en cuenta que la totalidad de estos términos y otros similares deben asociarse a las magnitudes físicas adecuadas y son meramente etiquetas convenientes aplicadas a estas magnitudes. A no ser que se establezca específicamente lo contrario según se manifieste a partir de la siguiente argumentación, debe apreciarse que en la totalidad de la descripción, las argumentaciones que utilizan términos tales como “computación”, “comparación”, “aplicación”, “creación”, “ordenación”, “clasificación” o similares, se refieren a las acciones y procesos de un sistema de ordenador, o dispositivo informático electrónico similar, que manipula y transforma datos representados en forma de magnitudes físicas (por ejemplo, electrónicas) dentro de los registros y memorias del sistema de ordenador para obtener otros datos representados de manera similar como magnitudes físicas dentro de las memorias o registros del sistema de ordenador u otros dispositivos de este tipo de almacenamiento, transmisión o visualización de información.

Ciertas formas de realización de la presente invención se refieren también a un aparato para llevar a cabo las operaciones de la presente memoria. Este aparato se puede construir con los fines pretendidos, o puede comprender un ordenador de propósito general activado o reconfigurado de manera selectiva por un programa de ordenador almacenado en el ordenador. Dicho programa de ordenador se puede almacenar en un soporte de almacenamiento legible por ordenador, tal como, aunque sin carácter limitativo, cualquier tipo de disco incluidos discos flexibles, discos ópticos, CD-ROM y discos magnetoópticos, memorias de solo lectura (ROM), memorias de acceso aleatorio (RAM), EPROM, EEPROM, tarjetas magnéticas u ópticas, o cualquier tipo de soportes adecuado para almacenar instrucciones electrónicas.

Debe entenderse que la descripción anterior está destinada a ser ilustrativa, y no limitativa. A los expertos en la materia les resultarán evidentes muchas otras formas de realización al leer y entender la descripción anterior. El alcance de la invención queda definido por las reivindicaciones adjuntas.

REIVINDICACIONES

1. Método implementado por ordenador, que comprende:

recibir, por parte de un procesador (402), un mapeo de datos que describe una asociación entre uno o más campos de una ubicación de almacenamiento de datos de una fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) y uno o más campos de una ubicación de almacenamiento de datos de un destino objetivo (102A, 102B, 102C, 104A, 104B, 104C);

analizar por lo menos uno de los datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C), metadatos que describen los datos de la fuente de datos, y características de almacenamiento lógicas y físicas asociadas a los datos de la fuente de datos, para detectar dinámicamente particiones asociadas a dichos datos, de manera que dichas particiones comprenden particiones físicas y/o lógicas;

determinar un grado de paralelismo que se va a utilizar cuando se transfieren los datos entre la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) y el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) basándose en las particiones detectadas dinámicamente,

generar, por parte del procesador (402), un plan de ejecución de transferencia de datos a partir del mapeo de datos para transferir datos desde la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C), comprendiendo el plan de ejecución de transferencia de datos el grado determinado de paralelismo que se va a utilizar cuando se transfieren los datos desde la ubicación de almacenamiento de datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta la ubicación de almacenamiento de datos del destino objetivo (102A, 102B, 102C, 104A, 104B, 104C); y

transferir, por parte del procesador (402), los datos desde la ubicación de almacenamiento de datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta la ubicación de almacenamiento de datos del destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) usando el plan de ejecución de transferencia de datos generado.

2. Método según la reivindicación 1, que comprende asimismo:

recibir información que describe la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C), comprendiendo la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) los datos que se van a transferir al destino objetivo (102A, 102B, 102C, 104A, 104B, 104C); y

recibir información que describe el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C), destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) que va a recibir los datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C).

3. Método según la reivindicación 1, que comprende asimismo:

recibir una solicitud para transferir los datos desde la ubicación de almacenamiento de datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta la ubicación de almacenamiento de datos del destino objetivo (102A, 102B, 102C, 104A, 104B, 104C).

4. Método según la reivindicación 1, que comprende asimismo:

proporcionar un manifiesto que comprende información que describe un resultado producido cuando se transfieren los datos desde la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C).

5. Método según la reivindicación 1, en el que los datos se transfieren desde la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) en paralelo de acuerdo con el grado de paralelismo determinado.

6. Sistema (400), que comprende:

una memoria (404); y

un procesador (402) acoplado a la memoria (404) para:

recibir un mapeo de datos que describe una asociación entre uno o más campos de una ubicación de almacenamiento de datos de una fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) y uno o más campos de una ubicación de almacenamiento de datos de un destino objetivo (102A, 102B, 102C, 104A, 104B, 104C);

analizar por lo menos uno de entre los datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C), los metadatos que describen los datos de la fuente de datos, y las características de almacenamiento lógicas y físicas asociadas a los datos de la fuente de datos, para detectar dinámicamente particiones asociadas a dichos datos, de manera que dichas particiones comprenden particiones físicas y/o lógicas;

determinar un grado de paralelismo que se va a utilizar cuando se transfieren los datos entre la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) y el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) basándose en las particiones detectadas dinámicamente;

generar un plan de ejecución de transferencia de datos a partir del mapeo de datos para transferir datos desde la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C), comprendiendo el plan de ejecución de transferencia de datos el grado determinado de paralelismo que se va a utilizar cuando se transfieren los datos desde la ubicación de almacenamiento de datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta la ubicación de almacenamiento de datos del destino objetivo (102A, 102B, 102C, 104A, 104B, 104C); y

transferir los datos desde la ubicación de almacenamiento de datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta la ubicación de almacenamiento de datos del destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) usando el plan de ejecución de transferencia de datos generado.

7. Sistema según la reivindicación 6, en el que el procesador (402) es asimismo para:

recibir información que describe la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C), comprendiendo la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) los datos que se van a transferir al destino objetivo (102A, 102B, 102C, 104A, 104B, 104C); y

recibir información que describe el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C), destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) que va a recibir los datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C).

8. Sistema según la reivindicación 6, en el que el procesador (402) es asimismo para:

recibir una solicitud para transferir los datos desde la ubicación de almacenamiento de datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta la ubicación de almacenamiento de datos del destino objetivo (102A, 102B, 102C, 104A, 104B, 104C).

9. Sistema según la reivindicación 6, en el que el procesador (402) es asimismo para:

proporcionar un manifiesto que comprende información que describe un resultado producido cuando se transfieren los datos desde la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C).

10. Soporte legible por ordenador (428) que tiene instrucciones grabadas en el mismo que, cuando son ejecutadas por un procesador (402), provocan que el procesador (402) lleve a cabo operaciones que comprenden:

recibir, por parte del procesador (402), un mapeo de datos que describe una asociación entre uno o más campos de una ubicación de almacenamiento de datos de una fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) y uno o más campos de una ubicación de almacenamiento de datos de un destino objetivo (102A, 102B, 102C, 104A, 104B, 104C);

analizar por lo menos uno de entre los datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C), los metadatos que describen los datos de la fuente de datos, y las características de almacenamiento lógicas y físicas asociadas a los datos de la fuente de datos, para detectar dinámicamente particiones asociadas a dichos datos, de manera que dichas particiones comprenden particiones físicas y/o lógicas;

determinar un grado de paralelismo que se va a utilizar cuando se transfieren los datos entre la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) y el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) basándose en las particiones detectadas dinámicamente;

generar, por parte del procesador (402), un plan de ejecución de transferencia de datos a partir del mapeo de datos para transferir datos desde la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C), comprendiendo el plan de ejecución de transferencia de datos el grado determinado de paralelismo que se va a utilizar cuando se transfieren los datos desde la ubicación de almacenamiento de datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta la ubicación de almacenamiento de datos del destino objetivo (102A, 102B, 102C, 104A, 104B, 104C); y

transferir, por parte del procesador (402), los datos desde la ubicación de almacenamiento de datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta la ubicación de almacenamiento de datos del destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) usando el plan de ejecución de transferencia de datos generado.

11. Soporte legible por ordenador de la reivindicación 10, que comprende asimismo por lo menos uno de entre:

recibir información que describe la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C), comprendiendo la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) los datos que se van a transferir al destino objetivo (102A, 102B, 102C, 104A, 104B, 104C), y recibir información que describe el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C), destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) que va a recibir los datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C),

recibir una solicitud para transferir los datos desde la ubicación de almacenamiento de datos de la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta la ubicación de almacenamiento de datos del destino objetivo (102A, 102B, 102C, 104A, 104B, 104C),

proporcionar un manifiesto que comprende información que describe un resultado producido cuando se transfieren los datos desde la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C), y

los datos son transferidos desde la fuente de datos (102A, 102B, 102C, 104A, 104B, 104C) hasta el destino objetivo (102A, 102B, 102C, 104A, 104B, 104C) en paralelo de acuerdo con el grado de paralelismo determinado.

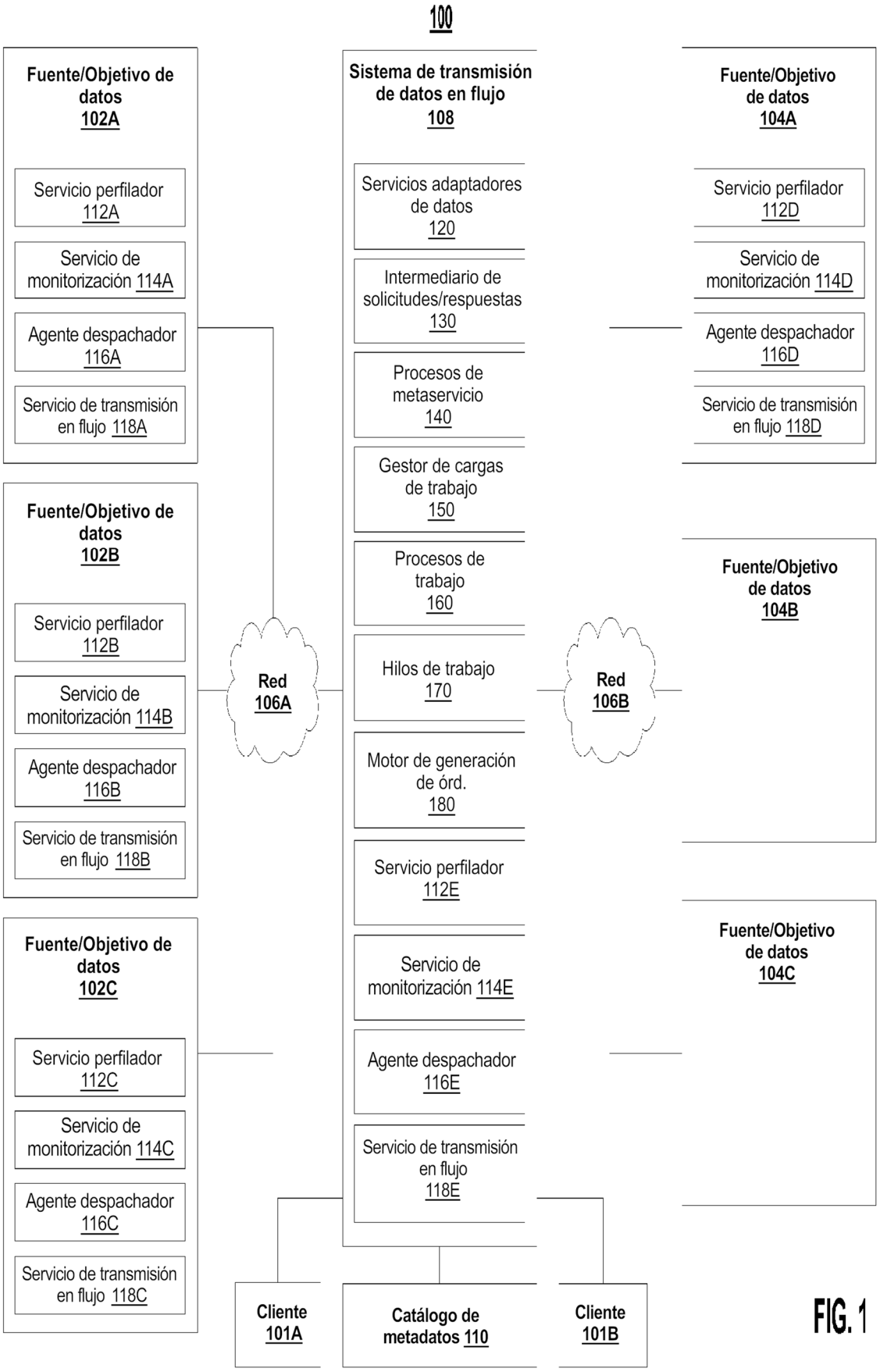
100
Fuente/Objetivo de datos 102A
Servicio perfilador 112A
Servicio de monitorización 114A
Agente despachador 116A
Servicio de transmisión en flujo 118A
Fuente/Objetivo de datos 102B
Servicio perfilador 112B
Servicio de monitorización 114B
Agente despachador 116B
Servicio de transmisión en flujo 118B
Fuente/Objetivo de datos 102C
Servicio perfilador 112C
Servicio de monitorización 114C
Agente despachador 116C
Servicio de transmisión en flujo 118C
Red 106A
Sistema de transmisión de datos en flujo 108
Servicios adaptadores de datos 120
Intermediario de solicitudes/respuestas 130
Procesos de metaservicio 140
Gestor de cargas de trabajo 150
Procesos de trabajo 160
Hilos de trabajo 170
Motor de generación de órd. 180
Servicio perfilador 112E
Servicio de monitorización 114E
Agente despachador 116E
Servicio de transmisión en flujo 118E
Red 106B
Fuente/Objetivo de datos 104A
Servicio perfilador 112D
Servicio de monitorización 114D
Agente despachador 116D
Servicio de transmisión en flujo 118D
Fuente/Objetivo de datos 104B
Fuente/Objetivo de datos 104C
Cliente 101A
Catálogo de metadatos 110
Cliente 101B

FIG. 1

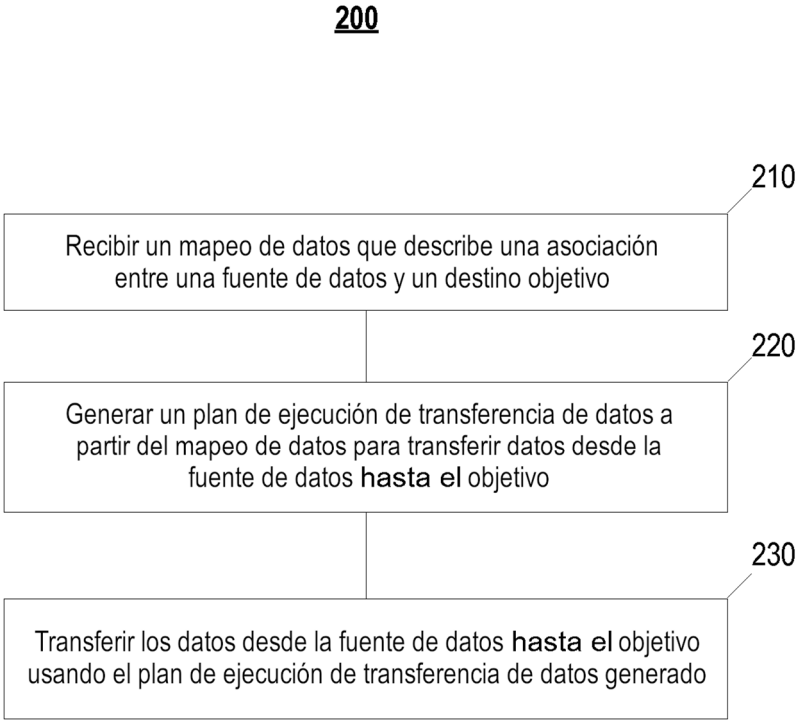
200
210
Recibir un mapeo de datos que describe una asociación entre una fuente de datos y un destino objetivo
220
Generar un plan de ejecución de transferencia de datos a partir del mapeo de datos para transferir datos desde la fuente de datos hasta el objetivo
230
Transferir los datos desde la fuente de datos hasta el objetivo usando el plan de ejecución de transferencia de datos generado

FIG. 2

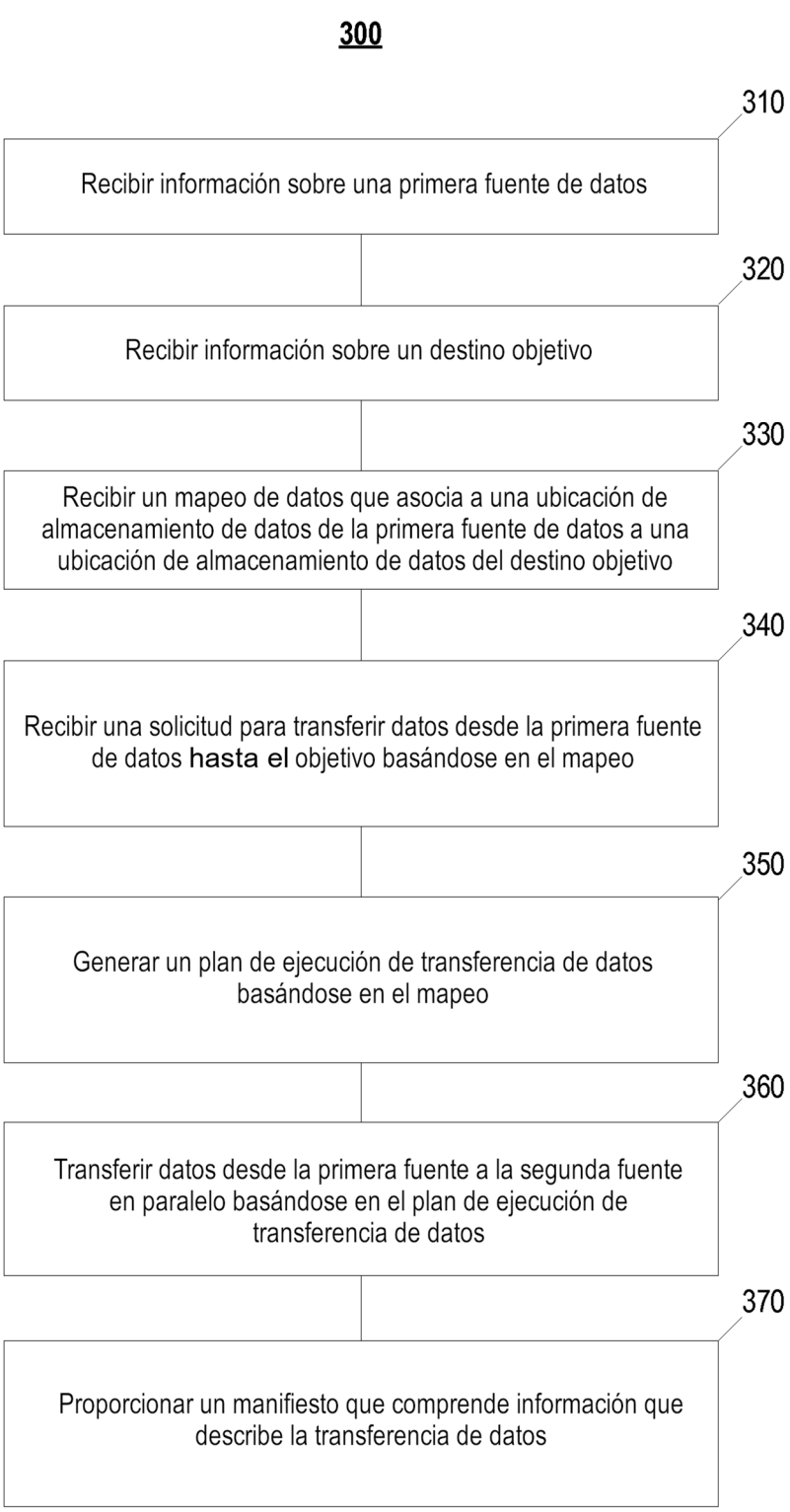
300
310
Recibir información sobre una primera fuente de datos
320
Recibir información sobre un destino objetivo
330
Recibir un mapeo de datos que asocia a una ubicación de almacenamiento de datos de la primera fuente de datos a una ubicación de almacenamiento de datos del destino objetivo
340
Recibir una solicitud para transferir datos desde la primera fuente de datos hasta el objetivo basándose en el mapeo
350
Generar un plan de ejecución de transferencia de datos basándose en el mapeo
360
Transferir datos desde la primera fuente a la segunda fuente en paralelo basándose en el plan de ejecución de transferencia de datos
370
Proporcionar un manifiesto que comprende información que describe la transferencia de datos

FIG. 3

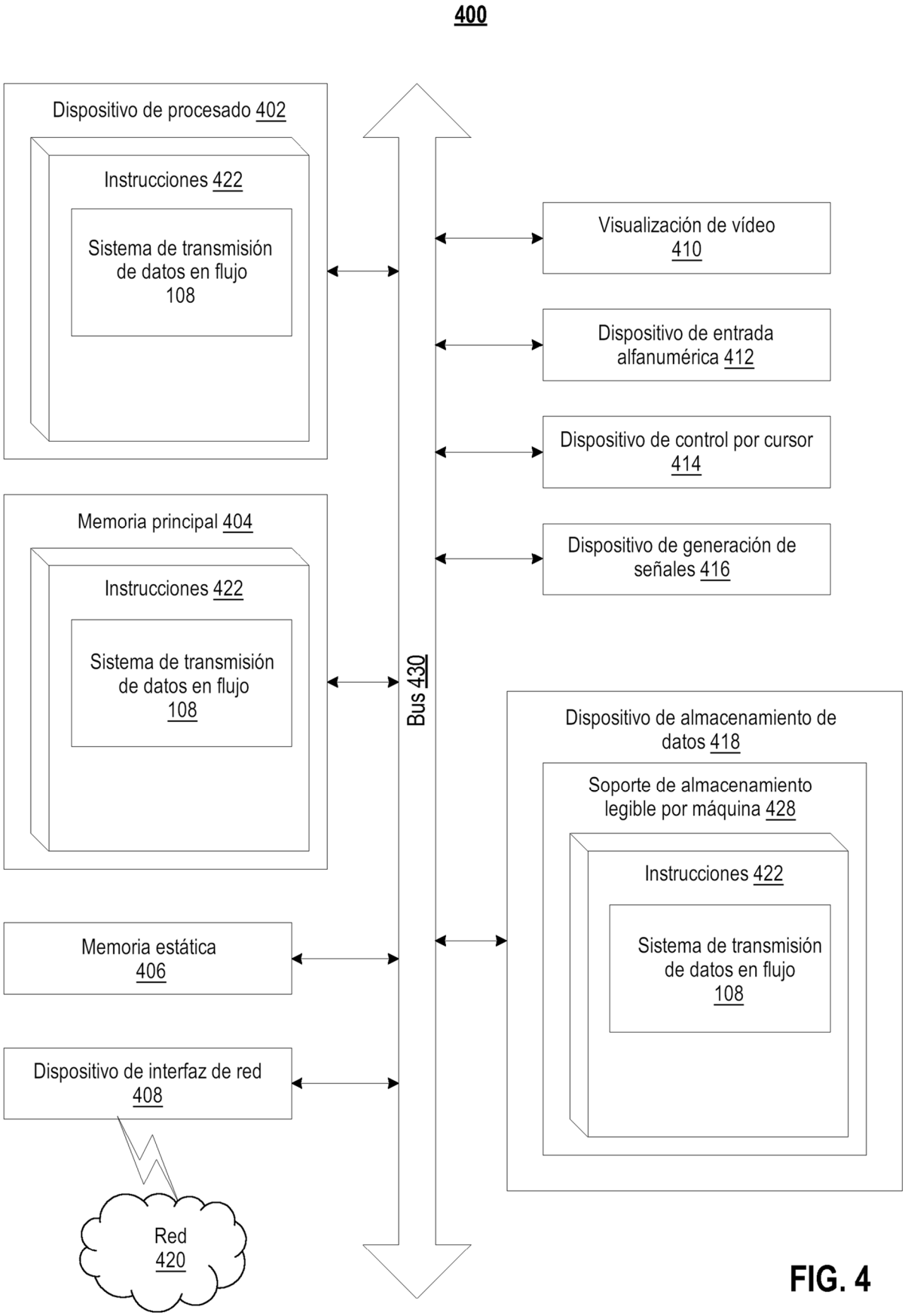
400
Dispositivo de procesado 402
Instrucciones 422
Sistema de transmisión de datos en flujo 108
Memoria principal 404
Instrucciones 422
Sistema de transmisión de datos en flujo 108
Memoria estática 406
Dispositivo de interfaz de red 408
Red 420
Bus 430
Visualización de vídeo 410
Dispositivo de entrada alfanumérica 412
Dispositivo de control por cursor 414
Dispositivo de generación de señales 416
Dispositivo de almacenamiento de datos 418
Soporte de almacenamiento legible por máquina 428
Instrucciones 422
Sistema de transmisión de datos en flujo 108

FIG. 4