(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 374 388**

(51) Int. CI.:
G06F 3/06 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Número de solicitud europea: **01906769 .3**
(96) Fecha de presentación: **29.01.2001**
(97) Número de publicación de la solicitud: **1256050**
(97) Fecha de publicación de la solicitud: **13.11.2002**

(54) Título: **MECANISMO DE TRANSFERENCIA DE DATOS DE ALTA VELOCIDAD.**

(30) Prioridad:
01.02.2000 US 495751

(45) Fecha de publicación de la mención BOPI:
16.02.2012

(45) Fecha de la publicación del folleto de la patente:
16.02.2012

(73) Titular/es:
COMMVAULT SYSTEMS, INC.
2 CRESCENT PLACE
OCEANPORT, NJ 07757-0090, US

(72) Inventor/es:
IGNATIUS, Paul;
PRAHLAD, Anand;
DEVASSEY, Varghese;
TYAGARAJAN, Mahesh;
WU, Robert y
IYER, Shankar, I.

(74) Agente: **Carpintero López, Mario**

DESCRIPCIÓN

Mecanismo de transferencia de datos de alta velocidad

Solicitudes relacionadas

La presente solicitud es una continuación en parte de la Solicitud de Patente de los Estados Unidos Nº de serie 09/038.440, presentada el 11 de Marzo de 1998, que se basa en la Solicitud Provisional de los Estados Unidos Nº 60/063.831, presentada el 30 de Octubre de 1997.

Campo de la invención

La invención se refiere a mecanismos de transferencia de datos y, en particular, a una Conducción de Datos "DataPipe" de alta velocidad, basada en software para proporcionar una transferencia de datos fiable y de alta velocidad entre ordenadores.

Antecedentes de la invención

Es bastante obvio que los datos, en el procedimiento de archivarse o transferirse desde una localización a otra, pasarán a través de diversas fases donde tendrán lugar diferentes operaciones tales como la compresión, transferencia de red, almacenamiento, etc. sobre los mismos. Hay esencialmente dos enfoques que pueden hacerse cuando se implementa tal mecanismo de transferencia. Uno sería dividir el procedimiento de archivo en sub-tareas, cada una de las cuales realizaría una función específica (por ejemplo, la Compresión). Esto requeriría a continuación la copia de los datos dentro de las sub-tareas, lo que podría resultar en un uso intensivo del procesador. El otro método sería minimizar las copias, y tener un programa monolítico que realizaría todas las funciones de archivo. La desventaja de este método sería la pérdida de paralelismo. Una tercera alternativa sería, por supuesto, usar hilos (threads) para hacer estas tareas y usar protocolos de señalización de hilos, sin embargo, se ha constatado que esto no sería enteramente práctico ya que los hilos no se soportan totalmente sobre muchas plataformas de cálculo.

Por consiguiente, es altamente deseable obtener un mecanismo de transferencia de datos de alta velocidad implementado en software y desarrollado para las necesidades de transferencia de datos fiable y de alta velocidad entre ordenadores.

Es un objeto de la invención desvelar la implementación de la Conducción de Datos de acuerdo con el producto de recuperación y respaldo de Vault98 del Sistema CommVault. Durante el desarrollo de la Conducción de Datos, se asume que los datos, a medida que se mueven desde la fuente de archivo (cliente de respaldo) al destino de archivo (servidor de respaldo en oposición a los medios), pueden sufrir una transformación o examen en diversas etapas entre ellos. Esta puede ser acomodar diversas acciones tales como la compresión de datos, indexación, envoltura de objetos, etc. que necesitan realizarse sobre los datos que se están archivando. Otra suposición es que los datos pueden transmitirse sobre la red a las máquinas remotas o transferirse a medios conectados localmente para su archivo.

Tanto los ordenadores de transmisión como de recepción ejecutan un software denominado en este documento como la Conducción de Datos. Aunque el mecanismo de transferencia de la Conducción de Datos a describir en este documento es un componente clave de los esquemas de los productos software de respaldo y recuperación, la Conducción de Datos no está restringida a este uso. Es un mecanismo de transferencia de datos de propósito general implementado en software que es capaz de mover los datos sobre una red entre un ordenador transmisor y uno receptor a muy altas velocidades y en un modo que permite la plena utilización de uno o más trayectos de red y la plena utilización del ancho de banda de la red. También puede usarse una Conducción de Datos para mover los datos desde un dispositivo de almacenamiento a otro dentro de un ordenador único sin el uso de una red. De este modo, el concepto de la Conducción de Datos no está confinado a su implementación sólo en sistemas conectados en red, sino que se puede operar también para transferir datos en ordenadores no conectados en red.

El documento WO 98/39707 desvela un sistema y un método para monitorizar descargas por una aplicación cliente. Estos se describen de modo que están particularmente adaptados para su uso con transferencias en modo de flujos. Sin embargo, el sistema de la técnica anterior no considera si el modo de transferencia de los datos es el mejor adaptado para la operación de almacenamiento. En particular, este documento desvela un sistema de almacenamiento de datos que tiene un dispositivo de almacenamiento y motores de datos de la fuente y de destino. El sistema usa un protocolo de transferencia de ficheros (FTP) para el almacenamiento de un fichero, y, en un modo de transferencia, el fichero se divide en bloques estando cada uno de los bloques precedido por uno o más bytes de cabecera.

Sumario de la invención

Es un objeto de la invención proporcionar en un sistema de comunicaciones que tiene un dispositivo de almacenamiento de origen y un dispositivo de almacenamiento de destino, un aparato de conducción de transferencia de datos para transferir datos en una secuencia de N etapas, donde N es un número entero positivo mayor que 1, desde el dispositivo de almacenamiento origen al de destino.

De acuerdo con la presente invención se proporciona un sistema de almacenamiento de datos como se define en la reivindicación 1.

En realizaciones de la invención, el aparato puede comprender memoria dedicada que tiene un número predeterminado de memorias intermedias dedicadas para transportar los datos asociados con la transferencia de datos desde el dispositivo o procedimiento de origen al dispositivo o procedimiento de destino; y un módulo de control maestro para registrar y controlar los procedimientos asociados con el aparato de transferencia de datos para la participación en las N etapas de la secuencia de transferencia de datos. Los procedimientos incluyen al menos una primera etapa de procedimiento para la inicialización de la transferencia de datos y una última etapa de procedimiento de orden N para completar la transferencia de datos. La primera etapa de procedimiento es operativa para asignar una memoria intermedia del número predeterminado de memorias intermedias disponibles dentro de la memoria dedicada para la recogida, procesamiento y envío de los datos desde el dispositivo de origen a la siguiente etapa de procedimiento.

La última etapa de procedimiento de orden N es operativa para recibir una memoria intermedia asignada a la primera etapa de procedimiento desde la etapa de orden (N - 1) de procedimiento en la secuencia de transferencia de datos y para liberar la memoria intermedia una vez que se ha completado el procesamiento y el almacenamiento en el dispositivo de destino para permitir la reasignación de la memoria intermedia. El procedimiento de control maestro incluye además un medio para la monitorización de las diversas memorias intermedias de la pila de memorias intermedias adjudicadas o asignadas a procedimientos particulares en la conducción, en el que el medio de monitor se puede operar para impedir la asignación de memorias intermedias adicionales a procedimientos particulares cuando el número de memorias intermedias asignadas actualmente excede un umbral predeterminado.

Descripción de los dibujos

La invención se entenderá mejor con referencia a los siguientes dibujos, en los que:

La FIG. 1 es un diagrama de bloques de una arquitectura de la Conducción de Datos.
La FIG. 2A es un esquema del procedimiento de transferencia de la Conducción de Datos sobre un único ordenador.
La FIG. 2B es un esquema del procedimiento de transferencia de la Conducción de Datos sobre múltiples ordenadores.
La FIG. 2C es un esquema del procedimiento de asignación de memorias intermedias de transferencia de la Conducción de Datos desde una pila de memorias intermedias almacenadas en una memoria compartida.
La FIG. 2D es un esquema que ilustra el control de las relaciones de un procedimiento monitor maestro a diversos procedimientos adjuntos.
Las FIG. 3A – 3C ilustran diversos mensajes transferidos entre procedimientos de aplicación y el procedimiento monitor maestro.
Las FIG. 4A – 4B ilustran esquemas de un procedimiento de módulo adjunto para el espacio de memoria compartido.
Las FIG. 5A – 5B representan diagramas de flujo de la operación de los procedimientos del secuenciador y re-secuenciador.
La FIG. 6 representa un flujo de transferencia de datos de ejemplo entre diversas etapas de procesamiento dentro de la Conducción de Datos.
La FIG. 7 ilustra un procedimiento de transferencia de la conducción de datos sobre ordenadores múltiples que tienen procedimientos con instancias múltiples.
La FIG. 8 representa un sistema modular de datos y gestión de almacenamiento que funciona de acuerdo con los principios de la presente invención.
La FIG. 9 es una cabecera de ejemplo que está típicamente colocada al comienzo de los fragmentos de datos que se envían a través del sistema de gestión de almacenamiento de la FIG. 8.
La FIG. 10 es un sistema modular de datos y gestión de almacenamiento de ejemplo.
La FIG. 11 es una realización de ejemplo de otro sistema modular de datos y gestión de almacenamiento.
La FIG. 12 es una realización de ejemplo de una configuración funcional para un almacenamiento de la información de cabecera.
La FIG. 13 es una realización de ejemplo de otra configuración funcional potencial para el movimiento de datos entre un motor de los datos fuente y un motor de los datos de destino.

Descripción detallada

Antes de embarcarnos en una descripción detallada del mecanismo de transferencia de datos de la presente invención, debería entenderse lo siguiente. El objetivo de la Conducción de Datos descrita es mover datos tan rápidamente como sea posible desde el punto A al punto B (que pueden estar sobre el mismo o diferentes ordenadores dentro de una red) mientras que se realizan una diversidad de operaciones (compresión, cifrado, análisis de contenidos, etc.) sobre los datos. Para cumplir este objetivo, debe explotarse completamente el procesamiento en paralelo, debe utilizarse completamente el ancho de banda de la red, y los ciclos de la CPU deben minimizarse. La Conducción de Datos debe implementarse de forma eficaz sobre una amplia diversidad de sistemas de ordenador de modo que sistemas heterogéneos sobre una red puedan usar una Conducción de Datos para

transferir datos entre sí.

Una Conducción de Datos comprende un conjunto nombrado de tareas que se ejecutan dentro de uno o más ordenadores que cooperan entre sí para transferir y procesar datos en un modo segmentado. Dentro de una Conducción de Datos, se usa un concepto de conducción para mejorar el funcionamiento de la transferencia de datos a través de múltiples ordenadores en una red. Sin embargo, dentro de una Conducción de Datos, cualquier etapa dentro de la conducción puede tener múltiples instancias, aumentando de este modo enormemente la escalabilidad y actuación del concepto básico de la conducción.

El mecanismo de Conducción de Datos procesa los datos dividiendo su procesamiento en tareas lógicas que pueden realizarse en paralelo. A continuación secuencia esas tareas en el orden en que están para actuar sobre los datos. Por ejemplo, una tarea de cabecera puede extraer datos desde una base de datos, una segunda tarea puede cifrarlos, una tercera puede comprimirlos, una cuarta puede enviarlos sobre la red, una quinta puede recibirlos desde la red, y una sexta puede escribirlos a una cinta. Las últimas dos tareas pueden residir sobre un ordenador diferente que las otras, por ejemplo.

Todas las tareas que comprende una única Conducción de Datos sobre un ordenador determinado tienen acceso a un segmento de memoria compartida que está dividida en varias memorias intermedias. Se usa un pequeño conjunto de primitivas de manipulación de las memorias intermedias para asignar, liberar, y transferir las memorias intermedias entre tareas.

Se usan semáforos (u otras primitivas específicas OS de exclusión mutua o señalización) para coordinar el acceso a las memorias intermedias entre tareas sobre un ordenador determinado. Tareas especiales, llamadas agentes de red, envían y reciben los datos a través de conexiones de red usando protocolos normalizados de red. Estos agentes posibilitan a una Conducción de Datos conectar a través de múltiples sistemas de ordenadores. Una Conducción de Datos única puede residir por lo tanto sobre más de un ordenador y podría residir sobre ordenadores de diferentes tipos.

Cada una de las tareas puede implementarse como un hilo, proceso, o procedimiento separado dependiendo de las capacidades del sistema de computación sobre el que está implementada la Conducción de Datos.

El paradigma del intercambio de datos llamado Conducción de Datos se ha puesto de moda para proporcionar soluciones a los problemas asociados y encontrados en los sistemas de transferencia de datos de la técnica anterior. Las características sobresalientes de este método son las siguientes:

1. Divide la tarea global del procesamiento de datos en sub tareas lógicas y las secuencia de acuerdo con el orden en el que se supone que actúan sobre el flujo de datos.
2. Usa procedimientos / hilos dedicados para realizar la transferencia de red.
3. Hace que todas las tareas dedicadas compartan un único gran segmento de memoria compartida.
4. Divide el segmento de memoria compartida en pequeñas memorias intermedias de modo que este espacio único de memoria intermedia puede compartirse entre diversos hilos de ejecución en las diversas etapas de las tareas.
5. Usa semáforos (u otras primitivas específicas OS de exclusión mutua o de señalización) para transferir el control sobre los segmentos de datos entre los módulos.

Como se ha mencionado anteriormente, cada una de las tareas puede implementarse como un hilo, proceso o procedimiento separado en un procedimiento monolítico (en casos en los que las plataformas nativas no soportan ninguna forma de ejecución en paralelo o multi procesamiento). Para la transferencia de datos a través de la red, las lecturas y escrituras de red dedicadas aseguran la comunicación a través de la red. La FIG. 1 muestra un cuadro del estado estable de cómo se establece la arquitectura de la Conducción de Datos 10 de acuerdo con la presente invención.

Refiriéndonos a la FIG 1, se muestra un disco 20 que reside sobre una máquina de ordenador 30, tal como una SUN MICROSYSTEMS INC., SPARCSTATION 2, que aloja la información o los datos a respaldar o archivar en el ordenador servidor 40 (que puede ser por ejemplo un SPARC 10) a través de los dispositivos ópticos o DLT 50 y 60 respectivamente. Como se puede comprobar, la Conducción de Datos representa la arquitectura extremo a extremo que puede utilizarse durante el respaldo de la base de datos desde el controlador de disco 20 donde se archivará la base de datos a la cinta o dispositivos ópticos 50 y 60 en el servidor 40. De este modo la Conducción de Datos elimina la red como factor limitante en el funcionamiento de respaldo. Como resultado, la pila de dispositivos define las capacidades de funcionamiento.

Como se muestra en la FIG. 1, la Conducción de Datos o flujo 70 se crea para la transferencia de datos para cada uno de los dispositivos en la pila de dispositivos a usar simultáneamente, que comprende los módulos 72, 74, 76, 78, 79 y 50. De forma similar, se muestra una segunda Conducción de Datos 80 comprendida por los módulos 82, 84, 76, 78, 89 y 60. Obsérvese que si se usan dispositivos de archivo adicionales para los datos de respaldo y en paralelo, se proporcionarían Conducciones de Datos adicionales. Como se puede comprobar, el concepto de Conducción de Datos a través de la explicación de un trayecto o hilo por el que se transfieren los datos, la descripción adicional se centrará en el procesamiento a través de una única Conducción de Datos o flujo 70, como

se muestra en la FIG. 1. En la cabecera de la Conducción de Datos está el componente de recogida 72 que es el responsable de la obtención de la información de la base de datos desde el disco 20. Los datos se pasan hacia abajo en memorias intermedias que residen en una memoria dedicada compartida (por ejemplo, una memoria RAM) a través de la conducción 70, a través de un módulo de compresión opcional 74, para los módulos de interfaz de red 76. En la interfaz de red, se multiplexan los datos y las trayectorias de red en paralelo 77 obtiéndose una tasa de transferencia máxima a través de la red. La red puede ser, por ejemplo, la bien conocida Ethernet, o cualquier red capaz de soportar los protocolos TCP/IP incluyendo FDDI o las redes ATM. El número de trayectos de red utilizados para cada uno de los flujos es un parámetro configurable determinado por el ancho de banda de la red y configurable a través de una interfaz de usuario. Obsérvese que a medida que son necesarios niveles más altos de funcionamiento, pueden usarse dispositivos adicionales simultáneamente con las interfaces de red adicionales añadidas y utilizadas para aumentar adicionalmente la tasa transferencia de la red. Del lado de recepción, desde el servidor de la base de datos 40, el esfuerzo del dispositivo parece local para la máquina y la arquitectura de la Conducción de Datos parece como una nube sin restricciones para el funcionamiento. El módulo de interfaz de red 78 opera para transferir los datos recibidos a través de la red al dispositivo 50 para su almacenamiento en el servidor 40. De este modo, la tarea final de almacenamiento o archivo de los datos se realiza en el módulo del dispositivo de archivo 50.

De la descripción anterior y la Figura 2A, se puede comprobar que la conducción o la Conducción de Datos 10 comprende una tarea de cabecera 15 que genera los datos a archivar o transferir desde el almacén 50, y una tarea de cola 40 que cumplen la tarea final de almacenamiento o escritura de los datos al almacén 60, incluyendo el archivo o restablecimiento de los datos. Pueden existir uno o más módulos intermedios 20, 30, que procesan los datos realizando acciones tales como la compresión, cifrado, análisis de contenidos, etc. asignando o sin asignar nuevas memorias intermedias mientras que se realiza el procesamiento.

Puede disponerse una conducción sobre una máquina particular para proporcionar una alimentación a otra máquina diferente. Un diagrama esquemático se ilustra en la Figura 2B. En este caso, la Conducción de Datos reside sobre más de un ordenador. Esto se hace con la ayuda de los agentes de red y los procesadores de control 50A, 50B, 60A y 60B. En tales casos, la primera máquina 12A tiene una cabecera 15 y otros módulos 20, 30, etc. que comprenden procedimientos intermedios. Un grupo de agentes de red dedicados 50A que envían datos a través de la máquina remota 12B mediante los protocolos de red normalizados y actúan como una seudo cola sobre la primera máquina. Sobre la máquina remota, un grupo de agentes de lectura de red dedicados 50B actúan como una seudo cabecera, y junto con otros módulos tales como los intermedios (no mostrados) y el de cola 70, constituyen el segmento de la conducción sobre esa máquina.

Además de la transferencia de datos desde un ordenador a otro, una capacidad única de la invención de la conducción es la capacidad de escalar para posibilitar la utilización total del ancho de banda de una red, y utilizar completamente varios dispositivos periféricos tales como los controladores de cinta, o utilizar completamente otros componentes hardware tales como las CPU. La escalabilidad de una Conducción de Datos se consigue usando múltiples instancias a cada una de las tareas de la conducción.

Por ejemplo, múltiples tareas de cabecera que operan en paralelo pueden recoger los datos desde una base de datos y depositarlos dentro de las memorias intermedias. Estas memorias intermedias pueden procesarse a continuación por varias tareas en paralelo que realizan una función tal como el cifrado. Las tareas de cifrado a su vez pueden alimentar varias tareas en paralelo para realizar la compresión y varias tareas en paralelo pueden realizar las operaciones de transmisión de red para explotar completamente el ancho de banda de la red. Sobre el ordenador objetivo, varias tareas del lector de red pueden recibir datos, que se escriben a múltiples unidades de cinta por varias tareas. Todas estas tareas sobre ambos ordenadores son parte de la misma Conducción de Datos y realizan colectivamente el trabajo de mover los datos desde la base de datos a las unidades de cinta. Hacen este trabajo extremadamente eficaz por la utilización completa de todo el ancho de banda disponible y el hardware asignado a la Conducción de Datos mientras que también minimizan los ciclos de CPU evitando la copia innecesaria de los datos a medida que se mueven de una etapa de la Conducción de Datos a la siguiente.

La FIG. 2B muestra el caso de múltiples ordenadores donde una única tarea de cabecera (procedimiento de recogida) recoge los datos desde el disco 40 y los deposita dentro de las memorias intermedias. Las memorias intermedias se procesan a continuación por varias instancias en paralelo del procedimiento de compresión 20 que una vez que se completa el procesamiento de cada una de las memorias intermedias para cada una de las instancias envía la memoria intermedia de procedimiento, al procedimiento 30 que realiza el análisis del contenido, y envía los datos procesados de la memoria intermedia a varias tareas de agentes de red 50A o instancias, que realizan las operaciones de red para enviar los datos sobre la red física 55 donde se reciben y se procesan por los agentes de red correspondientes 50B sobre el ordenador remoto 12B y envían al procedimiento de respaldo/restauración de cola 70 para almacenamiento o escritura al controlador de DLT 80.

En general podría haber N etapas en una conducción o Conducción de Datos determinada. En cada una de las etapas de la conducción podría haber p instancias de una tarea de módulo determinado. Estas N etapas podrían estar todas sobre la máquina local o podrían dividirse a través de dos máquinas diferentes en cuyo caso hay escritores de red y lectores de red (es decir, agentes de red de seudo cola y cabecera) que trabajan juntos para asegurar la continuidad en la conducción.

Refiriéndonos a la FIG. 2B, cada una de las Conducciones de Datos tiene un segmento de memoria dedicado 85 sobre cada una de las máquinas sobre la cual reside la Conducción de Datos. Por ejemplo, una Conducción de Datos que envía datos desde la máquina 12A a la máquina 12B tiene dos segmentos de memoria dedicados, uno sobre la máquina A y otro sobre la máquina B. Las tareas que son parte de esta Conducción de Datos pueden asignar y liberar memorias intermedias entre estos segmentos de memoria. Por supuesto, las tareas que operan sobre la máquina 12A sólo pueden asignar o liberar memorias intermedias dentro del segmento de memoria 85 sobre la máquina A y del mismo modo para tareas sobre la máquina B. De este modo, cualquiera de estos módulos puede asignar o liberar elementos de una única gran memoria compartida sobre cada una de las máquinas dedicadas para su uso por esta conducción particular.

Primitivas de manipulación de las memorias intermedias

Refiriéndonos ahora a la FIG. 2C, cada una de las tareas o procedimientos (15) que desea asignar una memoria intermedia lo hace de una pila de memorias intermedias 75 almacenadas en el segmento de memoria compartida 85 perteneciente a la Conducción de Datos usando la función AsigMem (). Cada una de las tareas que desea procesar los datos entrantes desde la tarea anterior ejecuta una llamada de recepción usando la función RecibirMem (). Cda una de las tareas que desea abandonar el control de una memoria intermedia particular de modo que la siguiente tarea pueda operar sobre la misma, realiza una función EnviarMem () sobre esa memoria intermedia para enviarla a la siguiente tarea. Cada una de las tareas que desea destruir una memoria intermedia y devolverla a la pila de memorias intermedias lo hace ejecutando una función LiberarMem () sobre esa memoria intermedia.

El Monitor_Maestro está conectado a un puerto predeterminado, para posibilitarle la comunicación con sus pares sobre otros sistemas de comunicaciones. El Monitor_Maestro monitoriza el estado de todas las Conducciones de Datos bajo su control en todo momento y puede proporcionar el estado de la Conducción de Datos a la aplicación software que usa la Conducción de Datos.

Para el cumplimiento de las tareas anteriores, un programa gestor del maestro llamado Monitor_Maestro se ejecuta en la realización preferida como un demonio sobre todas las máquinas de procedimientos. El programa de Monitor_Maestro "oye" o recibe los datos de la señal de control sobre un puerto dedicado para recibir tales datos de control de los procedimientos externos. De ese modo, el programa Monitor_Maestro puede servir los requisitos de las operaciones de la conducción. El Monitor_Maestro funciona para monitorizar el estatus de todas las conducciones bajo su control en todo momento y reporta el estatus de la conducción a todos sus sub-módulos. Como se muestra en las Figuras 2B y 2D, el Monitor_Maestro incluye tomas de mensajería de control abiertas a todos los módulos a través de los cuales puede controlar o cambiar el estatus de ejecución de cada uno de los módulos. El Monitor_Maestro 90 incluye además funciones que monitorizan el estado y los listados de todos los recursos compartidos de forma central (de entre los diversos módulos de la misma conducción) tales como la memoria compartida o los semáforos o cualquier recurso similar. El Monitor_Maestro a menos que se solicite de otro modo iniciará todos los módulos de la conducción bien por la función tenedor () o el hilo crear () o un hilo de OS específico similar del mecanismo de control de iniciación. El Monitor_Maestro también permite la iniciación de una conducción con autenticación adecuada. Este procedimiento iniciador puede identificarse a sí mismo bien como un procedimiento de cabecera o un procedimiento de cola, que más tarde se unirá el mismo a la conducción. (Se hace la excepción en el caso de un módulo de trabajo en red, para esta facilidad. No se permitirá a un procedimiento de red adjuntarse a sí mismo como la cabecera o cola de cualquier conducción). El demonio Monitor_Maestro posee y controla la memoria de almacenamiento compartido 85 mostrada en la FIG. 2C y de este modo puede permitir o denegar los procedimientos de acceso a tal memoria.

Iniciación de la Conducción de Datos

Refiriéndonos ahora a la Figura 3A en conjunción con las FIG. 1 y 2A – D, se crea una Conducción de Datos llamando al Monitor_Maestro y pasándole un mensaje de Iniciar_Conducción. En este mensaje, parámetros tales como el nombre de la Conducción de Datos, los nombres de los módulos componentes de la Conducción de Datos, el número de instancias en paralelo para cada uno de los componentes, las propiedades de cada uno de los componentes (por ejemplo si asignan memorias intermedias o no), las máquinas locales y remotas involucradas en la Conducción de Datos, la dirección del flujo, la naturaleza del programa de invocación etc. se pasan al Monitor_Maestro. Obsérvese que el término "módulo" se refiere a un programa que se ejecuta como una tarea como parte de una instancia de una Conducción de Datos. Cada uno de los módulos puede tener más de una instancia (por ejemplo ejecutarse como más de una tarea) dentro de una Conducción de Datos.

Refiriéndonos ahora a la FIG. 3B, dependiendo de la naturaleza del programa de invocación, puede requerirse que el procedimiento que invoca la Conducción de Datos necesite identificarse por sí mismo al Monitor_Maestro local 90A y adjuntarse el mismo a la Conducción de Datos como una tarea de cabecera o de cola. Para operar sobre una red con dos ordenadores, el Monitor_Maestro 90 inicia un Procedimiento Controlador de la Red 60 sobre la primera máquina que contacta el Monitor_Maestro 90B sobre la segunda máquina donde esta Conducción de Datos se completará usando un mensaje de Extender Conducción. Toda la información requerida para el establecimiento del segundo lado de la Conducción de Datos (incluyendo el nombre de la Conducción de Datos, el número de procedimientos, el nombre de la máquina local y el número de instancias en paralelo de los procedimientos particulares) se pasan junto con esta llamada de modo que la Conducción de Datos se establece completamente a

través de ambas máquinas. En respuesta, el Monitor_Maestro 90B sobre la segunda máquina o máquina remota, inicia los procedimientos requeridos sobre la segunda máquina incluyendo el procedimiento de control de red 60B (véase la FIG. 2B) para iniciar los procedimientos del agente de red o máquina de recepción.

Identificación

El procedimiento responsable de la iniciación de la conducción construye un nombre para la conducción usando su propia Id de procedimiento, un sello temporal, y el nombre de la máquina donde está corriendo el procedimiento iniciador. Este nombre de la conducción se pasa junto con ambos mensajes de Iniciar-Conducción así como el EXTENDER_Conducción de modo que se identifica la conducción con el mismo nombre sobre todos los ordenadores en los cuales está operando (es decir, el remoto así como la máquina local). Todos los segmentos de memoria compartida y los semáforos (número de referencia 85 de la FIG. 2C) adjuntos a una conducción particular están referenciados por nombre con su nombre de conducción y sus compensaciones definidas. Por lo tanto el procedimiento de identificación de un semáforo específico o memoria compartida asociada con esta conducción es fácil y accesible para todos los procedimientos, y módulos ligados (es decir, módulos para los cuales el control se inicia por el Monitor_Maestro). Cada uno de los módulos ligados (es decir, un módulo no iniciado a través del Monitor_Maestro, que se adjunta por si mismo después de que se inicia la conducción) debe identificarse por si mimo a su Monitor_Maestro local a través de un mensaje ENVIAR_IDENTIDAD mostrado en la FIG. 3C. Este mensaje contiene el nombre de la conducción al que quiere adjuntarse por sí mismo el módulo ligado, una toma de control, y una id de procedimiento / hilo, que usa el Monitor_Maestro para monitorizar el estado de este módulo particular.

Implementación de la transferencia de datos

Asignación: Recepción: Enviar: Liberar

Dirigiendo la atención a la FIG. 2C y la FIG. 4, las memorias intermedias se asignan usando la llamada AsigMem (), desde una pila común de memorias intermedias especificada en la memoria compartida dedicada para la conducción particular. La pila consiste de un único gran espacio de memoria compartida 75 con el número de MaxMemorias de memorias intermedias igualmente dimensionadas y una estructura "req". La estructura "req" ilustrada en la FIG. 4 contiene colas de entrada y de salida para cada una de las etapas de la conducción sobre esa máquina particular. El acceso a la memoria compartida se controla usando un semáforo de lectura escritura.

Como se muestra en las FIG. 4A y B, la cola de entrada de la etapa de orden i de un módulo es la cola de salida de la etapa de orden (i – 1) del módulo. La cola de entrada del primer módulo es la cola de salida del último módulo de la conducción sobre esa máquina. La asignación de memorias intermedias siempre se realiza a partir de la cola de entrada asociada con la primera etapa del primer módulo o procedimiento y el primer conjunto de semáforos 50A – D están asociados cada uno únicamente con una cola particular para seguir el número de memorias intermedias asignadas por esa cola/módulo. Sin embargo, para asegurar que ninguna tarea de asignación puede consumir memorias intermedias injustamente, un segundo conjunto de semáforos 80 A – D está asociado cada uno de forma única con un módulo particular para limitar la asignación de memorias intermedias por cada uno de los módulos a un valor umbral de Max_Memorias / NA donde NA es el número de módulos del repartidor en la conducción sobre esta máquina particular. Estos parámetros se almacenan en la memoria 75 bajo el control del programa del Monitor_Maestro y determinan si cualquier procedimiento ha excedido su asignación. Esto significa que podría haber K memorias intermedias no liberadas en el sistema asignadas por una única instancia de un módulo H, donde K es Max_Memorias / NA. La asignación adicional por el módulo H será posible cuando una memoria intermedia asignada por H se libera.

Todas las llamadas a LiberarMem () liberan sus memorias intermedias en la cola de entrada del primer módulo. Por la misma regla, los módulos de la primera etapa no tienen nunca permitido hacer un RecibirMem () pero se les permite hacer AsigMem (). Por otra parte, los procedimientos de cola sólo tienen permitido realizar LiberarMem () y nunca tienen permitido realizar un EnviarMem (). Todos los módulos pueden Recibir, Asignar, Enviar, y Liberar memorias intermedias. Los módulos de la primera etapa siempre realizan EnviarMem () después de que ejecutan cada uno de AsigMem (). Nota: Cualquier módulo en la conducción puede asignar una memoria intermedia disponible si requiere copiar datos durante el procesamiento. Normalmente, sin embargo, los datos no se re-copian dentro de un segmento de conducción de la máquina determinada.

Como se ha mencionado anteriormente, cada una de las colas 95 está asociada con un semáforo 50 para garantizar el acceso ordenado a la memoria compartida y que pasa a activado bajo acciones tales como AsigMem (); RecibirMem (), EnviarMem () y LiberarMem (). Un segundo conjunto de semáforos 80, asociado cada uno con un módulo particular en la conducción proporciona un mecanismo para asegurar que no se producen cuellos de botella. Los agentes de red dedicados de este modo se mapean por si mismos a través de cualquier interfaz de red sobre el sistema, siempre que se asegure la propagación de los datos. El número de agentes de red por conducción es un parámetro configurable, que ayuda a este mecanismo a explotar el ancho de banda máximo de transferencia de datos disponible en la red sobre la que está operando. Un único hilo / procedimiento de la red padre dedicado monitoriza el funcionamiento y estatus de todos los agentes de red sobre esa máquina particular para una conducción particular.

Refiriéndonos de nuevo a la FIG. 4A, ahora se describe el flujo del procedimiento de la asignación, envío, recepción de memorias intermedias y la liberación de memorias intermedias por procedimientos en la conducción y sus índices de semáforos asociados. Una vez asignada una memoria intermedia por una primera etapa de módulo 15 a través del comando AsigMem (), el valor del semáforo 50A asociado con la cola 1 se disminuye desde el valor inicial S_0. Además, el segundo semáforo 80A que representa el índice del repartidor para este módulo particular (el módulo 15) que realiza la asignación también se disminuye de un valor inicial S_1. La Información que indica el módulo que realizó esta asignación está incluida dentro de cada una de las memorias intermedia. El módulo 15 envía a continuación la memoria intermedia a la cola 2 donde se recibe por el módulo 20 a través del comando RecibirMem (), sacado de la cola de entrada 2 y asignado al módulo que realizó la llamada (es decir el módulo 20). Una vez completado el procesamiento sobre esta memoria intermedia, el módulo 20 redirige la memoria intermedia por medio de EnviarMem () que redirige la memoria intermedia a la cola de destino (cola 3). El módulo 30 realiza a continuación un RecibirMem () de la memoria intermedia sobre su cola de entrada (es decir, la cola 3) y una vez procesados los datos, realiza la operación de LiberarMem (). Como parte de la operación de LiberarMem (), el semáforo 50A asociado con la cola 1 se aumenta. De forma similar, el semáforo 80A que es el índice del repartidor del módulo 15 (es decir el módulo que asignó esta memoria intermedia particular) también se aumenta. La información relevante para esta operación está siempre disponible con la memoria intermedia con la cual se está realizando la operación de liberar en virtud del área de memoria compartida 85. En la realización preferida, el primer conjunto de semáforos (50A – 50C) asociados con las colas de entrada/salida de una etapa particular pueden tener un valor umbral de hasta max_memorias que es indicativo del número máximo de memorias intermedias que pueden asignarse en la conducción. Sin embargo, los semáforos 80A – C asociados cada uno de forma única con un módulo particular de una etapa particular tiene un valor asociado de sólo max_memorias / NA, donde NA (número de repartidores) es mayor o igual que 1. Por consiguiente como el valor del semáforo para cualquiera de los semáforos 50A – C y 80A – C no puede ser menor que 0, esto asegura que los módulos del repartidor pueden obtener una partición justa del número total disponible de memorias intermedias de entrada.

La FIG. 4B ilustra la situación en la que al menos dos módulos son operables para asignar memorias intermedias. Refiriéndonos ahora a la FIG. 4B, que es similar a la FIG. 4A con la excepción de que los módulos 15 y 20 son ambos operables para asignar memorias intermedias, se describe ahora el siguiente procedimiento. El módulo 15 asigna la primera memoria intermedia y disminuye el semáforo 50A. De forma similar, el semáforo 80A también se disminuye. La memoria intermedia se envía a continuación mediante el comando enviar desde el módulo 15 de la cola 1 a la cola 2 donde el módulo 20 recibe la memoria intermedia y comienza el procesamiento. En este caso sin embargo, tal como durante la compresión, donde un módulo de compresión puede requerir la asignación de memorias intermedias adicionales para realizar su procesamiento, el módulo 20 realiza una Asignación () para asignar una nueva memoria intermedia de la pila de memorias intermedias disponibles en la memoria compartida 85. El funcionamiento de la Asignación por el módulo 20, causa de este modo que el semáforo 50A asociado con la cola 1, se disminuya adicionalmente. Además, el semáforo 80B asociado con el módulo 20 también se disminuye, ya que el módulo 20 es el que asigna la nueva memoria intermedia. Una vez procesada, la memoria intermedia original asignada por el módulo 15 se libera a través de la operación LiberarMem () del módulo 20 en la etapa 2 y el valor del semáforo 50A se aumenta en consecuencia. Además, el módulo 20 aumenta el semáforo 80A asociado con el módulo 15 como resultado del funcionamiento de la operación LiberarMem (), como se indica por la flecha 84. El módulo 20 realiza a continuación el EnviarMem () para enviar la nueva memoria intermedia al módulo 30 en la cola 3, donde el módulo 30 recibe a continuación la nueva memoria intermedia, realiza su procesamiento, y consecuentemente libera la memoria intermedia que aumenta el semáforo 50A, como se muestra por la flecha 86. Como parte de la operación LiberarMem (), el módulo 30 también aumenta el semáforo 80B asociado con el módulo 20 como se muestra por la flecha 88. De este modo, se impiden los cuellos de botella que se producen dentro de la conducción, mientras que se mantiene una tasa de transferencia de datos adecuada y eficaz.

Fijaciones

A medida que se completa el procedimiento de identificación, todos los módulos se fijan por si mismos a un segmento de memoria compartida específico que se comparte entre los módulos sobre esa máquina para esa conducción particular. Este segmento de memoria compartida tiene muchas memorias intermedias de datos, colas de entrada para todas las etapas sobre la conducción y sus valores iniciales. Cada uno de los módulos identifica sus propias colas de entrada y colas de salida dependiendo de la etapa en la que se supone que corre ese módulo, y la cola inicial (primera etapa) se puebla con varios segmentos de datos para compartir sobre esta conducción particular. También todos los módulos se acoplan por si mismos a una disposición de semáforos del repartidor, que controla el número de memorias intermedias asignadas por un módulo específico que puede activarse en la conducción.

Integridad de datos

La integridad de los datos pasados juntos y la secuenciación de los datos, se mantienen en parte por un par de módulos de propósito especial denominados procedimientos de secuenciador y re-secuenciador. Las Figuras 5A y 5B proporcionan diagramas del funcionamiento de los procedimientos de un secuenciador y un re-secuenciador respectivamente. Refiriéndonos a la Figura 5A, el procedimiento del secuenciador recibe cada una de las memorias intermedias (módulo 10), lee el número de secuencia actual almacenado en memoria (módulo 20), y a continuación sella la memoria intermedia con el número de secuencia actual (módulo 30) y envía la memoria intermedia sellada a

la siguiente etapa para su procesamiento (módulo 40). El número de secuencia actual se aumenta a continuación (módulo 50) y el procedimiento se repite para cada una de las memorias intermedias recibidas por el secuenciador. El re-secuenciador es operativo para recibir todas las memorias intermedias de entrada y almacenarlas internamente y esperar a que las memorias intermedias predecesoras requeridas aparezcan en la cola de entrada antes de redirigirlas todas en la siguiente secuencia a la siguiente etapa de procesamiento.

El propósito del re-secuenciador es forzar el ordenamiento adecuado de las memorias intermedias. Se asegura esto asegurando que envía memorias intermedias al siguiente módulo de la conducción en el orden del número de secuencia. Si se recibe una memoria intermedia fuera de orden, se mantiene por el re-secuenciador hasta que se reciben todas las memorias intermedias del procesador y se envían al siguiente módulo. De este modo, el ordenamiento de las memorias intermedias se garantiza y las memorias intermedias nunca se mantienen más tiempo que el necesario. Estas etapas se representan en la FIG. 5B. Obsérvese sin embargo, que cuando hay sólo una instancia de un módulo presente en cualquier etapa particular, en virtud del mecanismo de colas disponible con todas las colas de entrada, se asegura la secuencia de datos en el orden correcto.

Por lo tanto, en la realización preferida, todas las transferencias de datos de la conducción que emplean etapas multi-instancias a través de los procedimientos del secuenciador/re-secuenciador aseguran que la secuencia de entrada de los números de secuencia no se viola para cada una de las instancias del módulo. Además, la restricción de que todos los módulos de una etapa multi-instancia específica deben ser del mismo tipo elimina las posibilidades de comportamiento preferencial.

Justicia

El concepto de justicia significa que cada una de las tareas se asegurará de obtener las memorias intermedias de entrada que necesite para operar sobre ellas sin esperar más tiempo del necesario. La justicia entre los módulos en una Conducción de Datos determinada donde ninguna etapa de la conducción tiene más de una instancia es automática. A medida que la tarea de cola libera una memoria intermedia introduce la memoria intermedia liberada en la pila de memorias intermedias donde puede posibilitar a la tarea de cabecera su asignación y comenzar el procesamiento. Todas las tareas en la Conducción de Datos operan a la velocidad máxima solapando el procesamiento realizado por otras tareas en la etapa precedente o siguiente de la conducción.

Si una Conducción de Datos tiene etapas consistentes en instancias en paralelo de una tarea, la justicia entre esas tareas se asegura usando un semáforo de asignación que cuenta desde Max_Memorias / NA (donde NA es el número de repartidores para esta Conducción de Datos sobre esta máquina particular) hacia abajo hasta cero. Todas las LiberarMem () vuelven a aumentar este semáforo, sin embargo, podría haber sólo Max_Memorias / NA memorias intermedias asignadas por un módulo repartidor en esta Conducción de Datos. Esto asegura que todos los repartidores obtienen una justa partición del número total disponible de memorias intermedias de la entrada. Si un procedimiento particular intenta asignar más memorias intermedias que las que tiene permitidas, el procedimiento del Monitor_Maestro impide tal asignación, haciendo que el procedimiento termine o bien que espere hasta que una memoria intermedia asignada actualmente al procedimiento se libere aumentando por lo tanto el semáforo de respaldo para permitir el procedimiento de asignación de otra memoria intermedia.

Mensajes de control

Todas las instancias de todos los módulos tienen una toma de control para el Monitor_Maestro sobre el cual se intercambian mensajes de control. Todas las lecturas / escrituras tiene una toma de control análoga para su agente de la red padre. El propio agente de la red padre tiene una toma de control para el Monitor_Maestro. Cada uno de los módulos comprueba periódicamente su toma de control para cualquier mensaje desde el Monitor_Maestro. La información crítica tal como el mensaje PARAR_CONDUCCIÓN se pasa al Monitor_Maestro a través de este mecanismo.

Monitorización de estatus

Cada uno de los módulos iniciados por el Monitor_Maestro sobre una máquina determinada se monitoriza por cualquier procedimiento de la red padre (en el caso de una escritura o lectura de la red), o por el propio Monitor_Maestro, para estados de ejecución. En el caso de que se reporte por cualquier módulo que ha terminado anormalmente, el Monitor_Maestro identifica esta excepción, y señaliza a todos los módulos sobre esa conducción particular que paren. Esto se hace por medio de mensajes de control a través de las tomas de control como se ha descrito anteriormente. Una vez que se paran con seguridad todos los módulos pertenecientes a esta conducción particular, señaliza al Monitor Maestro de la máquina remota que pare el lado remoto de esta conducción particular y toda la conducción se cierra con seguridad por medio de la señalización de mensajes de control.

Implementación

En una realización preferida, las funciones de la Conducción de Datos y los procedimientos se implementan como una función software en el lenguaje de programa "C" de más alto nivel sobre sistemas operativos de Sun Solaris o de HP-UX y se incorporan dentro de la Revisión 2.7 del producto de gestión de almacenamiento Vault98 del Sistema CommVault.

La FIG. 6 es un ejemplo ilustrativo de la secuencia de comandos de primitivas, usados para constituir una Conducción de Datos. La Conducción de Datos se usa a continuación para procesar datos en tres módulos llamados A, B y C.

Para establecer la Conducción de Datos, se llama al Monitor_Maestro y se proporciona con el nombre de la Conducción de Datos y los nombres de los módulos que usarán la conducción (módulo 10).

El Monitor_Maestro (Iniciar_Conducción (Muestra_conducción, A, B, C)).

Dentro de la lógica del módulo A, se llama a continuación a la función AsigMem () para obtener una memoria intermedia (20). La lógica del módulo A puede realizar cualesquiera acciones que quiera para llenar la memoria intermedia con datos útiles. Cuando se ha completado su procesamiento de la memoria intermedia (30), llama a EnviarMem () para enviar la memoria intermedia al módulo B para su procesamiento (40). El módulo A repite a continuación su función de nuevo llamando a AsigMem () para obtener la siguiente memoria intermedia.

La lógica del módulo B llama a RecibirMem () para obtener una memoria intermedia de datos desde el módulo A (50). A continuación opera sobre la memoria intermedia realizando el procesamiento como se requiera (60). Cuando termina con la memoria intermedia llama a EnviarMem () para enviar esa memoria intermedia al módulo C (70).

El módulo B repite a continuación su función llamando de nuevo a RecibirMem () para obtener la siguiente memoria intermedia desde el módulo A.

El módulo C obtiene una memoria intermedia de datos a partir del módulo B llamando a RecibirMem (). Cuando se ha completado su procesamiento de datos en esa memoria intermedia (90), llama a LiberarMem () para liberar la memoria intermedia (100). Como los otros dos módulos, vuelve a recibir la siguiente memoria intermedia desde el módulo B.

Las primitivas usadas para asignar, liberar, enviar y recibir memorias intermedias se sincronizan por el uso de semáforos. Esto asegura la coordinación entre los módulos de modo que el módulo de recepción no comienza el procesamiento de datos antes de que el módulo de envío haya terminado con los mismos. Si no hay memoria intermedia disponible, las primitivas AsignarMem o RecibirMem esperarán hasta que una esté disponible. Todos los tres módulos operan en paralelo como tareas separadas. El orden de procesamiento desde A a B hasta C se establece en la llamada inicial al Monitor_Maestro que estableció la Conducción de Datos.

Refiriéndonos ahora a la FIG. 7, se muestra otra realización de la Conducción de Datos. El aparato de conducción se usa en Vault98 para proporcionar una trayectoria de alta velocidad entre un sistema "cliente" que contiene una gran base de datos que está respaldando el servidor "CommServ" y se almacena como ficheros de archivo sobre un controlador de DLT. Todos los elementos sobre el lado de recogida, de la red física son parte de la configuración de software del cliente, mientras que todos los elementos del lado del controlador de DLT de la red física son parte de la configuración software del servidor. Las actividades de "recoger" sobre el cliente preparan los datos a enviar sobre la conducción de Datos a CommServ.

La FIG. 7, que es similar a la FIG. 2B, representa una configuración de dos ordenadores donde la tarea de cabecera 15 identificada como un procedimiento de recogida, se inicia a través del demonio Monitor_Maestro 90A sobre el primer ordenador. El colector 15 recupera los datos desde el disco y asigna la memoria intermedia de la memoria compartida 85A para el procesamiento de los datos a transferir. El colector 15 envía a continuación los datos al procedimiento de compresión 20 que funciona para comprimir los datos a medida que se mueven sobre la conducción. Como se muestra en la FIG. 7, se proporcionan múltiples instancias del módulo de compresión 20 en esta etapa para el procesamiento de modo eficaz de los datos a medida que fluyen a través del sistema. Por consiguiente, el secuenciador 17 iniciado por el Monitor_Maestro 90A está acoplado directamente entre el módulo colector 15 y el módulo compresor 20 para sellar cada una de las memorias intermedias con el número de secuencia como se ha descrito anteriormente. El módulo re-secuenciador 23 está acoplado a la cola de salida de las instancias del módulo de compresión 20 para reordenar adecuadamente y re-secuenciar las memorias intermedias enviadas desde las instancias del módulo 20 al módulo de análisis de contenido 30. El módulo de análisis de contenido 30 recibe a continuación las memorias intermedias desde el re-secuenciador 23, procesa los datos, y envía las memorias intermedias al secuenciador 33, que sella de nuevo las memorias intermedias y las envía a múltiples instancias de los agentes de red 50A para el procesamiento a través de la red física mediante un protocolo normalizado de red tal como TCP IP, FTP, ICMP, etc. Los agentes de red 50B se instancian por el procesador de control de la red 60B en comunicación con el Monitor_Maestro remoto 90B para proporcionar múltiples instancias del agente de red, donde cada uno de los agentes sobre el lado remoto corresponde de forma única y comunica con el agente correspondiente sobre el lado local. En la realización preferida, cada uno de los agentes de red 50A del lado local realizan una copia de los datos en la memoria intermedia para transferir sobre la red física a su agente de red correspondiente 50B sobre el lado remoto y a continuación realiza una llamada a la función de liberar memoria intermedia para liberar las memorias intermedias asociadas con la memoria compartida 85A para su reasignación. Del lado remoto, el agente de red 50B recibe los datos transferidos sobre la red y actúa como una cabecera sobre el lado remoto para asignar cada una de las memorias intermedias en la memoria compartida 85B. Estas memorias intermedias se envían a continuación al re-secuenciador 53 que almacena las memorias intermedias recibidas en la

memoria interna hasta que se reciben cada una de las memorias intermedias del predecesor, y a continuación las redirige para el procedimiento de restaurar el respaldo 70 a través de la función EnviarMem (). El procedimiento de respaldo/restaurar funciona a continuación para escribir los contenidos de cada una de las memorias intermedias recibidas al controlador de DLT 80, y una vez completado, libera cada una de las memorias intermedias para permitir su reasignación adicional en la pila de memorias intermedias y la memoria compartida 85B. Como puede verse, esta conducción podría establecerse sobre cualquier red de alta velocidad, tal como ATM, FDDI, etc. La conducción es capaz de utilizar todo el ancho de banda práctico disponible sobre la red física por medio de múltiples agentes de red. En casos donde están disponibles las redes de alta velocidad real (redes que tienen tasas de transferencia más altas que los controladores de DLT), se establecen múltiples conducciones, para utilizar recursos disponibles para la extensión de la caída.

Características sobresalientes

De la discusión anterior, son evidentes numerosas ventajas del sistema de transferencia de datos de la conducción de datos usando memoria compartida señalizada por el uso de semáforos para producir un mecanismo de transferencia de datos flexible, de propósito general. Incluidas entre estas ventajas están:

1. Su naturaleza flexible – los módulos que se conectan dentro de una conducción pueden cambiarse fácilmente en base a la aplicación.
2. Permite tener múltiples instancias de un módulo determinado corriendo en una etapa determinada de la conducción. Esto permite un paralelismo sobre y más allá de lo que la conducción ya proporciona.
3. Proporciona un mecanismo bien definido para la puesta en marcha y cierre de una conducción e incluye mecanismos de gestión interna y limpieza proporcionados a través del Monitor_Maestro.
4. Permite el control de la aplicación sobre la cantidad del ancho de banda de la red del que quiere aprovecharse. Es posible fácilmente aprovechar completamente la ventaja de un mecanismo de transporte de ancho de banda simplemente aumentando el número de agentes de red.
5. Proporciona un esquema incorporado para la justicia entre módulos. En otras palabras, ningún módulo único puede retener todas las memorias intermedias de entrada, o ninguna instancia única de un módulo multi-etapa puede mantener otras instancias de operación.
6. Permite la fácil integración con el software de terceras partes en virtud del hecho de que la Conducción de Datos proporciona a cualquier módulo fijarse por sí mismo como un punto extremo sin consolidar (cabeza o cola).
7. Permite la fácil señalización de comprobación en virtud de una toma de conexión de cabeza-cola.

Sin embargo debería recordarse que la memoria compartida sobre una máquina particular no se comparte entre las otras diversas máquinas. De este modo, no estamos explotando resultados implícitos de una memoria compartida distribuida, sino haciendo una transferencia de datos, sólo en base a una demanda, descartando todas las memorias intermedias de maleza, con copia selectiva, para un mejor funcionamiento sobre el paradigma de la transferencia de datos. De este modo, la invención descrita en este documento representa un sistema de transferencia de datos real más que un paradigma de memoria compartida distribuida comúnmente visto.

La FIG. 8 es un sistema modular de datos y gestión de almacenamiento 800 que opera de acuerdo con los principios de la presente invención. Un primer sistema de operación 802 se ilustra que soporta una aplicación software 804 que se usa para almacenamiento y/o recuperación de datos. Para un fácil entendimiento de los principios de la presente invención, la FIG. 8 está ilustrada con datos que se están almacenando. Inicialmente los datos se mueven desde la aplicación software 804 a un motor de datos 806 donde se usa un módulo de mapeo de almacenamiento 808 para determinar donde se envían los datos y en qué formato. Los datos se envían a un motor de datos 810 antes de que se almacenen en el medio de almacenamiento 812.

El motor de datos 806 incluye un módulo de interfaz del sistema operativo 814 que interactúa con un módulo de cifrado 816, un módulo de compresión 818 y un gestor de fragmentación 820. El módulo de soporte de la cabecera/pie 821 se usa para registrar la información que indica en qué formato se han colocado los datos antes de moverse al motor de datos 810. El mapeo del almacenamiento 808 examina los datos para determinar donde se enviarán. En el ejemplo de la FIG. 8, los datos se envían al motor de datos 810 donde un módulo de interfaz con el medio 822 interactúa con los nuevos datos que se reciben en el motor de datos. El motor de datos 810 incluye un módulo de descifrado 824, un módulo de descompresión 826, un gestor de fragmentación 828, y un módulo de cabecera/pie 830. Uno cualquiera o más de estos componentes pueden activarse para alterar el formato de los datos que se han recibido en el motor de datos 810. El motor de datos 810 mueve a continuación los datos al medio de almacenamiento 812, tal como un primer medio de almacenamiento 832, un segundo medio de almacenamiento 834, o un medio de almacenamiento de orden n 836.

A medida que se mueven los datos desde el primer sistema operativo 802 al medio de almacenamiento 812, se crea una conducción de datos para el paso de los datos. Por ejemplo, el motor de datos 806 puede considerarse para crear una conducción de datos entre el motor de datos 806 y el motor de datos 810. Aunque los datos que se están transmitiendo podrían analizarse dentro de múltiples fragmentos por el gestor de fragmentación 820, y enviarse a los diferentes tipos de medios de almacenamiento 812, la conducción de datos puede considerarse que es la misma conducción de datos para los datos que se están enviando. Por ejemplo, si los datos que se envían desde el primer

sistema operativo 802 al medio de almacenamiento 812 son datos que comienzan en un formato de texto, cambian a un formato de video de reproducción directa, y a continuación a un formato de audio, los datos podrían separarse en fragmentos que deberían almacenarse en medios de almacenamiento diferentes y en diferentes formatos. Sin embargo, se considerará que los datos han viajado a través de una conducción de datos única. Cada uno de los fragmentos de datos que se envía al medio de almacenamiento 812 causa que el sistema de gestión de almacenamiento 800 identifique las características del fragmento que se ha enviado, así como las características del siguiente fragmento que se enviará, permitiendo por lo tanto al sistema de gestión de almacenamiento 800 mantener la conducción de datos que se ha establecido.

Cualquier porción del sistema de gestión de almacenamiento 800 puede seleccionar el formato para los datos. Por ejemplo, la aplicación de software 804 puede seleccionar si cifrar, comprimir o fragmentar los datos que se van a enviar. El mapeo de almacenamiento 808 también puede ser el componente que determine si cifrar, comprimir o fragmentar los datos. También, el motor de datos 806 puede tomar la decisión de si cifrar, comprimir o fragmentar los datos. Numerosas otras decisiones pueden hacerse por uno cualquiera de estos tres componentes, tal como el tipo de cabecera, el protocolo de transmisión, u otra información necesaria para transmitir datos desde el primer sistema operativo 802 al medio de almacenamiento 812.

Los motores de datos 806 y 810 se ilustran de modo que tienen otros módulos de soporte 838 y 840, respectivamente. Estos otros módulos de soporte 838 y 840, se ilustran con líneas discontinuas para indicar que pueden estar presentes o no.

La FIG. 9 es una cabecera de ejemplo 900 que típicamente se coloca al comienzo, de los fragmentos o datos que se envían a través del sistema de gestión de almacenamiento 800. La cabecera 900 incluye diversos elementos de información, tales como un control de versión 902. El control de versión 902 está sobre la estructura y ayuda en las versiones del formato de datos, de los datos que se están transmitiendo sobre el sistema de gestión de almacenamiento 800. Se incluye un módulo de compresión en línea 904 para asistir en la compresión de los datos en el tránsito de los datos desde una localización a la siguiente, y es una característica opcional.

Otra característica opcional es un módulo de vigilancia en línea 906 que se usa para tales propósitos como la comprobación de antivirus, así como otros propósitos de seguridad. Un módulo de transferencia de cabecera 908 está incluido para transferir cabeceras especiales con las porciones de datos que incluyen la cabecera 900. Se incluye un selector de algoritmos de compresión 910 para seleccionar el algoritmo de compresión apropiado para los datos que están a punto de enviarse o que se acaban de recibir. Un módulo de etiquetado de compensación y de bloques 912 se incluye para propósitos de etiquetar el número de bloque de compensación de datos. El número de bloque y la compensación de los datos es útil en la determinación de dónde localizar los datos que se han almacenado. También puede incluirse un generador de CRC en línea (comprobación de redundancia cíclica) 914 en la cabecera de etiqueta de orden i 900. Si el CRC en línea 914 está incluido, puede incluirse un módulo de descarte de CRC duplicados 916 para descartar bloques de CRC duplicados que se han generado por el generador de CRC en línea 914. También está incluido un módulo de re-arranque desde el punto de fallo (POF) 918 que es capaz de continuar la transmisión de datos independientemente de los fallos en la transmisión. También puede estar incluido un módulo de agrupación de bloques para agrupar juntos múltiples bloques de datos para una transferencia de datos más eficaz.

La FIG. 10 es en sistema modular de ejemplo de datos y gestión de almacenamiento 1000. El sistema de gestión de almacenamiento 1000 incluye sistemas de computación 1002, 1004, 1006 y 1008 que interactúan a través de una red 1010, tal como una red de ethernet. El sistema de computación 1002 incluye un primer sistema operativo 1012 que interactúa con la aplicación de software 1014. La aplicación de software 1014 puede ser una única aplicación o múltiples aplicaciones que interactúan con un sistema de ficheros instalado 1016. Cuando los datos se mueven desde el sistema de computación 1002, el sistema de ficheros instalados 1016 interactúa con un motor de datos 1018 que formatea los datos en una pluralidad de módulos 1020. El motor de datos 1018 hace interfaz con el sistema de computación 1008 para obtener información desde un gestor de almacenamiento 1022 concerniente a la localización del almacenamiento. Esta información está contenida en un almacenamiento maestro y un mapa de respaldo 1024. Una vez recibida la información apropiada desde el sistema de computación 1008, el motor de datos 1018 puede transmitir los datos al sistema de computación 1006 donde se reciben en el motor de datos 1026.

El motor de datos 1026 incluye una pluralidad de módulos 1028, un módulo de medios 1030, y un índice de datos 1032. La pluralidad de módulos 1028 permite al motor de datos 1026 almacenar los datos en una primer medio de almacenamiento 1034, un segundo medio de almacenamiento 1036, hasta un medio de almacenamiento de orden n 1038. Los datos que se envían desde el sistema de computación 1002 al sistema de computación 1006 pueden estar comprimidos múltiples veces antes de almacenarse en uno de los medios de almacenamiento 1034, 1036, 1038.

Además, el sistema de computación 1004 puede transmitir los datos a almacenar. El sistema de computación 1004 tiene un segundo sistema operativo 1040, aplicaciones de software 1042, un sistema de ficheros instalados 1044, y un motor de datos 1046 que tiene una pluralidad de módulos 1048. Como se describe en relación con la cabecera 900, los datos se transmiten en diversos formatos y diversas porciones del sistema de gestión de almacenamiento pueden determinar qué formatos implementar para la porción particular de la transmisión de datos.

Es de observar que los sistemas de computación 1002, 1004 y 1006 pueden incluir, como se muestra en las líneas discontinuas, los gestores de almacenamiento respectivos 1050, 1052 y 1054. En esta realización, el sistema de computación 1005 puede que ya no se requiera.

La FIG. 11 es una realización de ejemplo de otro sistema modular de datos y gestión de almacenamiento 1100. El sistema de almacenamiento y gestión 1100 incluye el sistema de computación 1102, el sistema de computación 1104, y el sistema de computación 1106. Los sistemas de computación 1102, 1104 y 1106 interactúan para almacenar los datos bien en un almacenamiento del área de red 1108 o en un almacenamiento conectado a la red 1110. La red 1112 está prevista para las comunicaciones con el almacenamiento conectado a la red 1110, mientras que la otra red 1114 (típicamente, una red de fibra de alta velocidad) está prevista para la comunicación con el almacenamiento del área de red 1108. Por ejemplo, el sistema de computación 1102 puede transmitir datos usando un primer sistema operativo 1116 que soporta aplicaciones software 1118 que interactúan con un sistema de ficheros instalado 1120 para transmitir datos a un motor de datos 1122. El motor de datos 1122 puede interactuar con un medio de almacenamiento 1124 para almacenar datos desde el sistema de computación 1102. El motor de datos 1122 también puede transmitir datos al motor de datos 1126 del almacenamiento del área de red 1108. Sin embargo, al tomar las decisiones de enviar datos al almacenamiento del área de red 1108, el sistema de conmutación 1106 está típicamente accedido para obtener información desde un módulo gestor 1144 para acceder al mapa maestro 1146 para la determinación de la localización de la transmisión de datos. Un módulo de medios 1128 del almacenamiento del área de red 1108 determina si los datos se almacenarán en el medio de disco magnético 1130, un medio óptico 1132 o un medio de cinta magnética 1134. Además el módulo de medios 1128 sigue la migración de los datos entre los diversos medios de almacenamiento 1130, 1132 y 1134.

El sistema de computación 1104 se ilustra incluyendo un segundo sistema operativo 1136, y aplicaciones de software 1138 que interactúan con un sistema de ficheros instalado 1140. El sistema de ficheros instalado 1140 puede recibir datos desde las aplicaciones de software 1138 y transmitir los datos al motor de datos 1142, donde se encuentra la información detallada concerniente a la transmisión de los datos en el sistema de computación 1106 y su módulo gestor 1144 y el mapa maestro 1146. Los datos se transmiten a continuación al almacenamiento conectado a la red 1110 donde un motor de datos de destino 1148 recibe los datos, y un módulo de medios 1150 determina donde se almacenarán los datos en un medio de almacenamiento 1152.

Como se muestra en las líneas discontinuas, la red 1112 podría extenderse directamente a la red del área de almacenamiento 1108. Como se muestra en líneas discontinuas, la red 1114 podría extenderse directamente al almacenamiento conectado a la red y al sistema de computación 1106. Estas variaciones crean una mayor flexibilidad en el sistema de gestión de almacenamiento 1100 y proporciona numerosas variaciones al sistema. Una vez vista la presente exposición, los especialistas en la técnica entenderán que son deseables numerosas variaciones en ciertas circunstancias.

La FIG. 12 es una realización de ejemplo de una configuración operativa para un almacenamiento de la información de cabecera 1211. El almacenamiento de la información de cabecera 1211 incluye un mapa de almacenamiento 1213, un índice de datos 1215 y "dentro de los fragmentos" 1217. Cada una de estas porciones del almacenamiento de la información de cabecera 1211 puede contener diferentes o todas las instrucciones para mover los datos desde un motor de datos fuente 1231 al motor de datos de destino 1233. Se ilustran algunos métodos de ejemplo para la transmisión de datos desde el motor de datos fuente 1231 al motor de datos de destino 1233. Por ejemplo, la cabecera1 1241 podría comenzar una transmisión de datos desde el motor de datos fuente 1231 al motor de datos de destino 1233. La cabecera1 1241 se seguiría por el fragmento1 1243. El fragmento1 1243 se seguiría a continuación por la cabecera2 1245. La cabecera2 1245 se seguiría por el fragmento2 1247, que a su vez se seguiría por una cabecera3 1249, que se seguiría por un fragmento3 1251, etc. De este modo, los datos se transfieren al motor de datos de destino 1233 en fragmentos hasta que se reciben los datos completos en el motor de datos de destino 1233. La configuración de cabeceras y fragmentos se controla por el almacenamiento de la información de cabeceras 1211. La información detallada para la transmisión de datos puede encontrarse en el mapa de almacenamiento 1213, el índice de datos 1215, y "dentro de los fragmentos" 1217 bien separadamente o conjuntamente.

Otro método para la transferencia de datos es donde una única cabecera 1261 comienza la transmisión de múltiples fragmentos, es decir fragmento1 1263, fragmento2 1265, fragmento3 1267, … fragmentoN 1269. Los fragmentos se siguen por un pie 1271 que completa la transmisión de los datos en esta realización particular.

Otro método más que puede usarse con el almacenamiento de la información de cabecera 1211 para la transmisión de datos se muestra por la cabecera1 1281, que se sigue por el fragmento1 1283. El fragmento1 1283 se sigue a continuación por un pie1 1285 para completar la transmisión de ese fragmento particular. El siguiente fragmento se envía del mismo modo, es decir, una cabecera2 1287 se sigue por un fragmento2 1289, que se sigue por un pie2 1291, para completar la transmisión de un segundo fragmento. Este procedimiento continúa hasta que todos los fragmentos se han transmitido al motor de datos de destino 1233. Por supuesto los tres métodos anteriores para la transmisión de datos son sólo de ejemplo, y podrían usarse otras alternativas para la transferencia de datos entre el motor de datos fuente 1231 y el motor de datos de destino 1233.

La FIG. 13 es una realización de ejemplo de otra configuración operativa potencial para mover datos entre un motor

de datos fuente 1310 y un motor de datos de destino 1312. En esta realización, se establece una conducción de datos y se envía una cabecera de sesión 1314 desde el motor de datos fuente 1310 al motor de datos de destino 1312 para indicar que debería establecerse una conducción entre los dos. Cuando está completa la conducción, se envía un pie de sesión 1316 desde el motor de datos fuente 1310 al motor de datos de destino 1312. Entre la cabecera de sesión 1314 y el pie de sesión 1316 están los archivos, es decir, la cabecera1 1318 seguida por el pie1 de archivo 1320, que se sigue por la cabecera2 de archivo 1322, que se cierra cuando se recibe el pie2 de archivo 1324, que continúa el procedimiento hasta que se recibe la cabeceraN de archivo 1326 y se recibe el pieN de archivo 1328 para establecer la terminación del archivo particular. Cada uno de los archivos está comprendido de fragmentos, como se ilustra por los fragmentos 1330, los fragmentos 1332 y los fragmentos 1334.

El fragmento 1330 se ilustra incluyendo el fragmento1 1336, el fragmento2 1338, … el fragmento N 1340. Cada uno de estos fragmentos individuales del fragmento 1330 se ilustra con mayor detalle y se representa por el fragmento 1350.

El fragmento 1350 incluye una cabecera de fragmento 1352 y el pie de fragmento 1354. La cabecera de fragmento 1352 se sigue por una cabecera de etiqueta 1356, que se sigue a continuación por los datos 1358. Otra cabecera de etiqueta 1360 sigue a los datos 1358, y se sigue por los datos 1362 y otra cabecera de etiqueta 1364. La cabecera de etiqueta 1364 se sigue por una cabecera de opciones 1366, que incluye información de procesamiento indicando tal información que los datos deberían almacenarse sobre un tipo diferente de medio de almacenamiento. La cabecera de opciones 1366 puede ser la única información que sigue la cabecera de etiqueta 1364, pero los datos 1368 se ilustran en el caso de que otros datos se incluyan después de la cabecera de datos 1364. Una cabecera de etiqueta 1370 se ilustra a continuación y se sigue por los datos 1372. Este procedimiento continúa hasta que el pie de fragmento 1354 se envía desde el motor de datos fuente 1310 al motor de datos de destino 1312.

Aunque se han mostrado las realizaciones preferentes de la presente invención, los especialistas en la técnica apreciarán además que la presente invención puede realizarse en otras formas específicas dentro del alcance de la presente invención como se define por las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un sistema de almacenamiento de datos (800) para la transferencia de un archivo de datos a al menos un dispositivo de almacenamiento (812), comprendiendo el sistema de almacenamiento de datos:

un motor de datos fuente (806), que:

divide el fichero en una pluralidad de fragmentos (1243, 1263, 1283) en base a los diferentes formatos de ficheros de datos dentro del fichero, y
asocia con cada uno de la pluralidad de fragmentos una información de cabecera (900) que contiene instrucciones respecto al fragmento asociado y el formato del fichero de los datos del fragmento asociado; y
un motor de datos de destino (810) que recibe la pluralidad de fragmentos y almacena la pluralidad de fragmentos en localizaciones sobre al menos un dispositivo de almacenamiento (812) de acuerdo con las instrucciones en la información de cabecera de cada uno de la pluralidad de fragmentos.

2. El sistema de almacenamiento de datos de la reivindicación 1, en el que cada uno de los fragmentos incluye una información de pie (1271, 1285) que instruye al motor de datos de destino (810) con respecto a los fragmentos.

3. El sistema de almacenamiento de datos de cualquiera de las reivindicaciones anteriores en el que la información de cabecera (900) comprende además al menos uno de los siguientes: control de versión (902), comprensión en línea (904), vigilancia en línea (906), transferencia de cabeceras (908), selección del algoritmo de compresión (910), etiquetado de compensación y bloque (912), comprobación de redundancia cíclica en línea (914), un módulo de descarte de la comprobación de redundancia cíclica redundante (916), un módulo de re-arranque desde el punto de fallo (918), y un módulo de bloqueo de grupo (920).

4. El sistema de almacenamiento de datos de la reivindicación 1, en el que el motor de datos fuente (806) analiza el fichero para determinar si enviar el fichero al motor de datos de destino (810) en fragmentos de acuerdo con el formato del fichero.

5. El sistema de almacenamiento de datos de acuerdo con cualquiera de las reivindicaciones anteriores en el que el motor de datos fuente (806) envía el fichero al motor de datos de destino (810) en fragmentos junto con la información de cabecera (900) que instruye al motor de datos de destino respecto a los fragmentos de acuerdo con el formato del fichero.

6. El sistema de almacenamiento de datos de acuerdo con cualquiera de las reivindicaciones anteriores en el que el formato de fichero comprende uno cualquiera o más del grupo consistente de formato de texto, formato de audio, y formato de video.

7. Un mecanismo de transferencia de datos de alta velocidad para el almacenamiento de datos que comprende un sistema de almacenamiento de datos de acuerdo con cualquier de las reivindicaciones anteriores y una aplicación de software (804) que puede usarse para generar los datos que se van a almacenar; en el que dicho motor de los datos fuente (810) recibe, desde la aplicación de software (804), los datos que se van a almacenar y en el que dicho al menos un dispositivo de almacenamiento (812) se usa para almacenar los datos que se generan por la aplicación de software.

8. Un mecanismo de transferencia de datos de alta velocidad para el almacenamiento de datos de la reivindicación 7, en el que el motor de datos fuente (806) configura los datos de modo que las instrucciones para el almacenamiento de los datos están contenidas tanto en la información de cabecera (900) como en al menos un pie (1271, 1285).

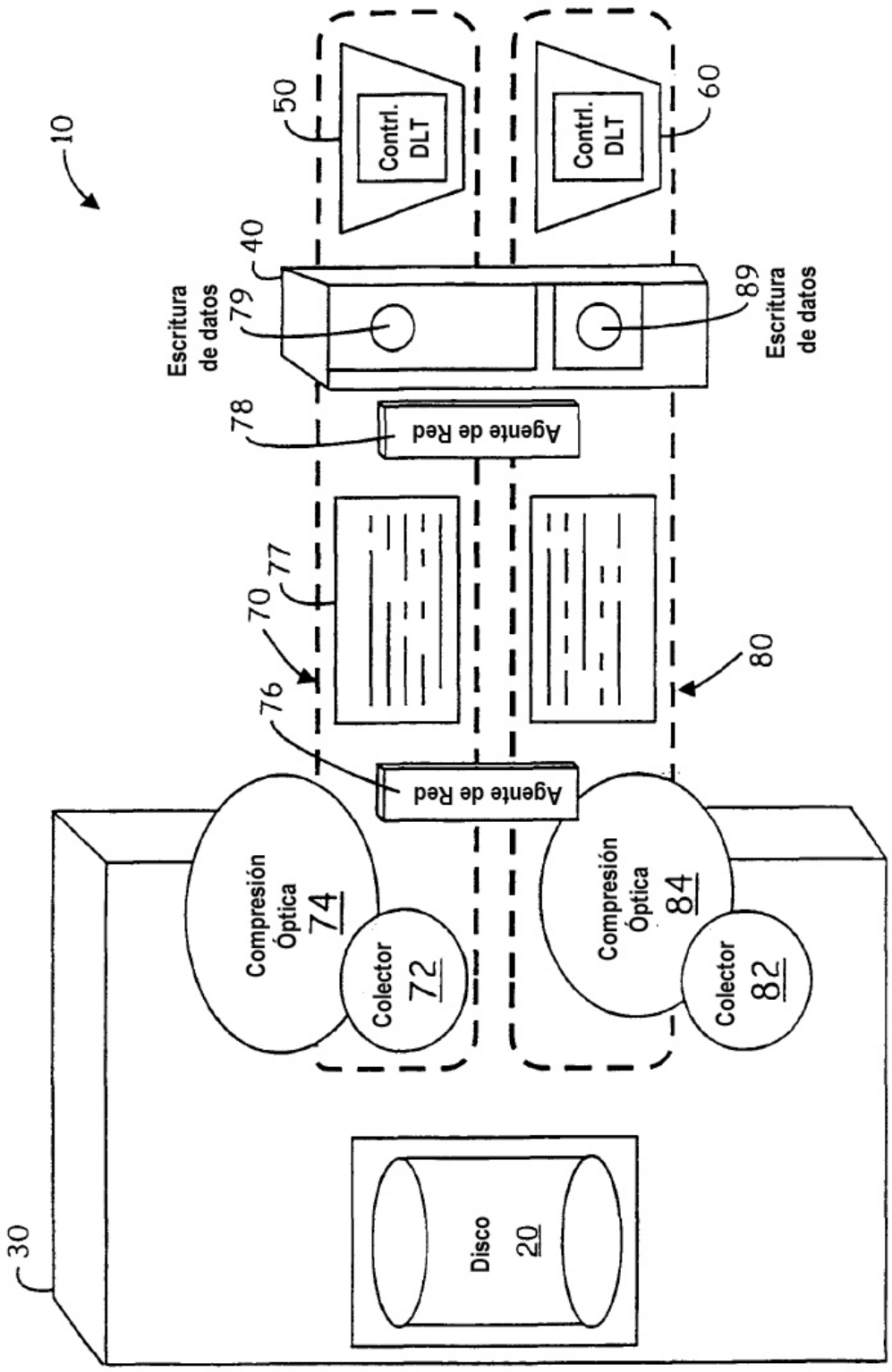
10
50
Contrl. DLT
60
Contrl. DLT
40
Escritura de datos
79
89
Escritura de datos
78
Agente de Red
77
70
80
76
Agente de Red
Compresión Óptica 74
Colector 72
Compresión Óptica 84
Colector 82
30
Disco 20

Fig. 1

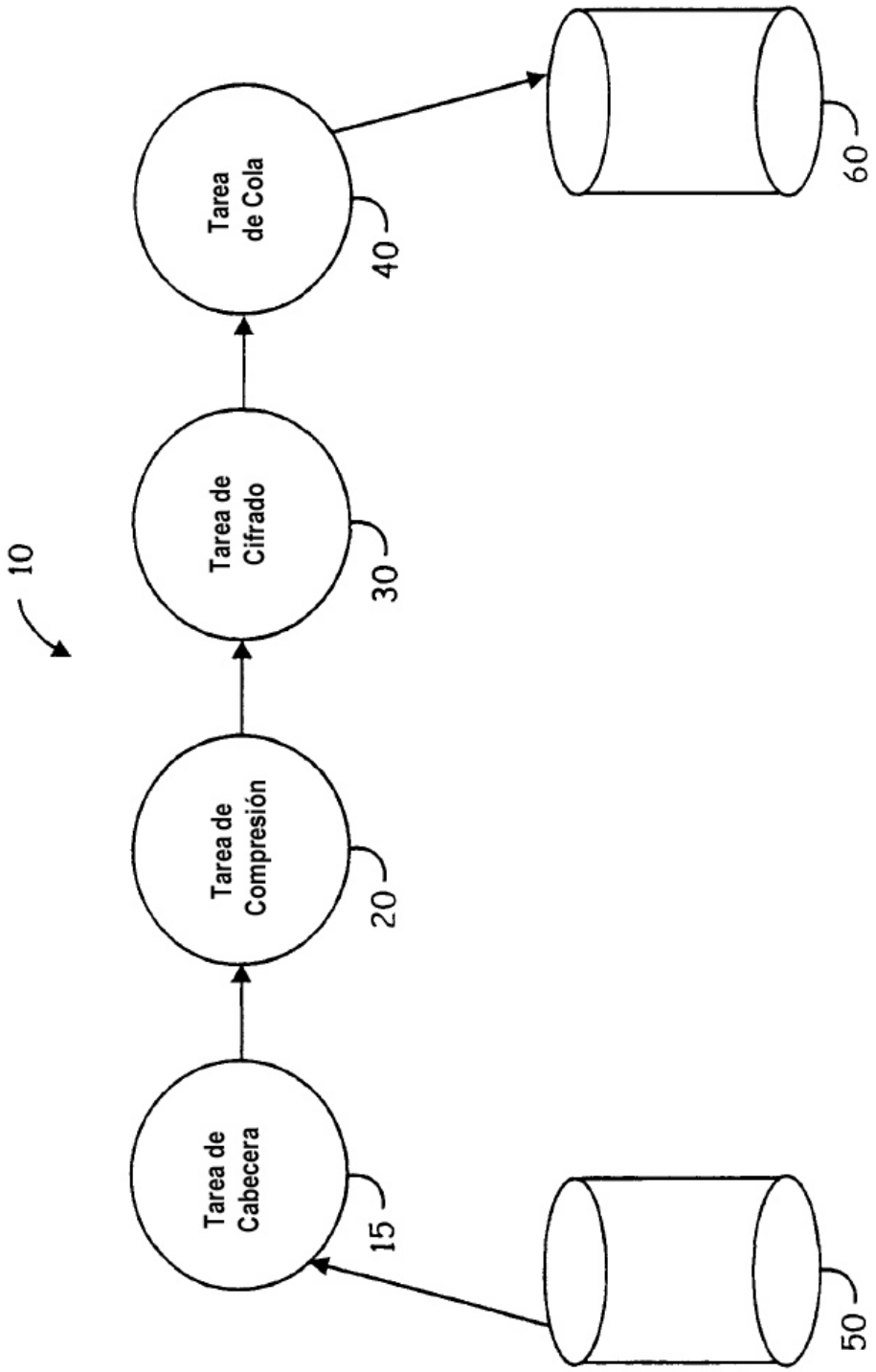
10
Tarea de Cabecera
15
Tarea de Compresión
20
Tarea de Cifrado
30
Tarea de Cola
40
50
60

Fig. 2A

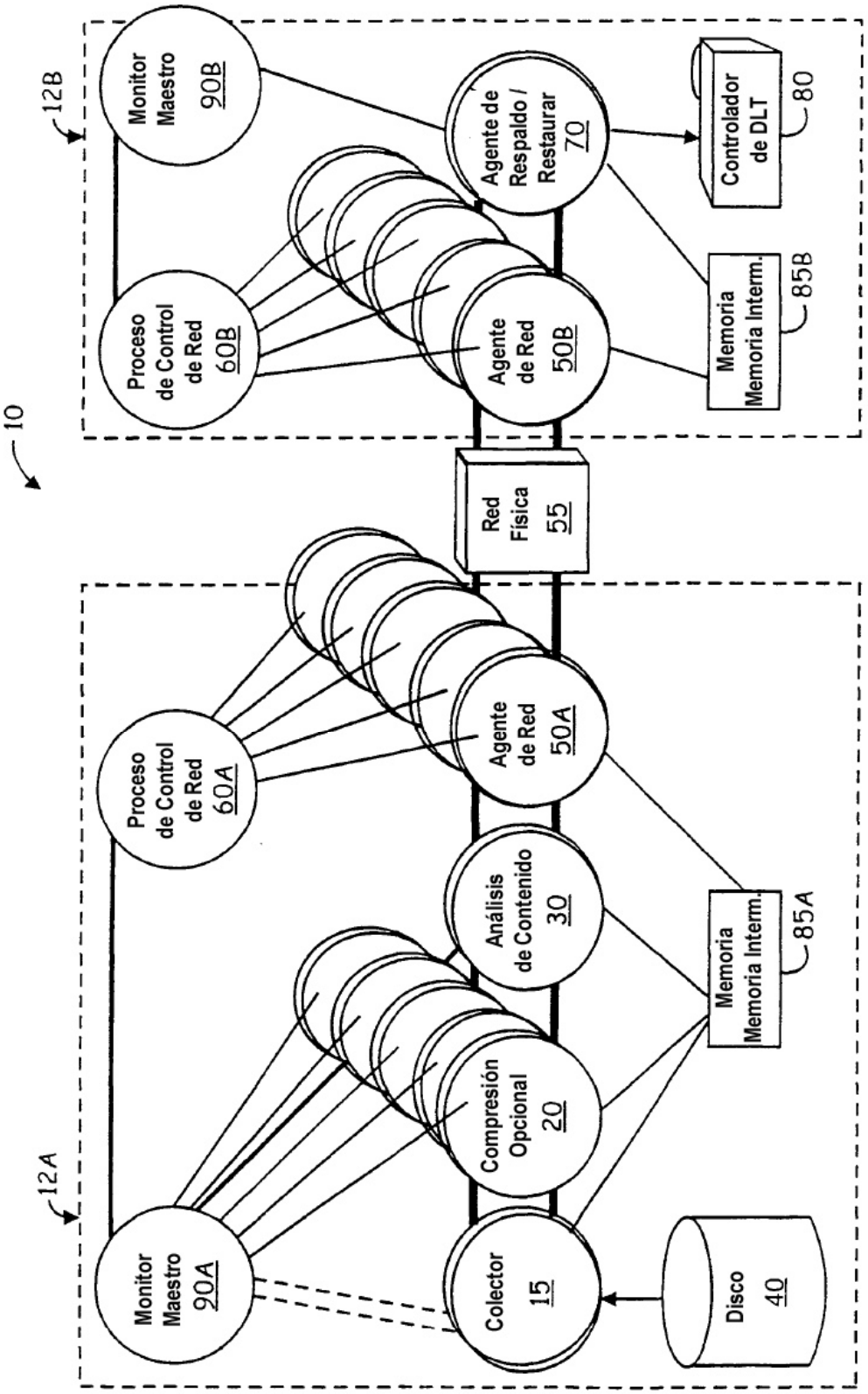
10
12B
Monitor Maestro 90B
Proceso de Control de Red 60B
Agente de Respaldo / Restaurar 70
Controlador de DLT 80
Agente de Red 50B
Memoria Memoria Interm. 85B
Red Física 55
12A
Proceso de Control de Red 60A
Agente de Red 50A
Análisis de Contenido 30
Memoria Memoria Interm. 85A
Monitor Maestro 90A
Compresión Opcional 20
Colector 15
Disco 40

Fig. 2B

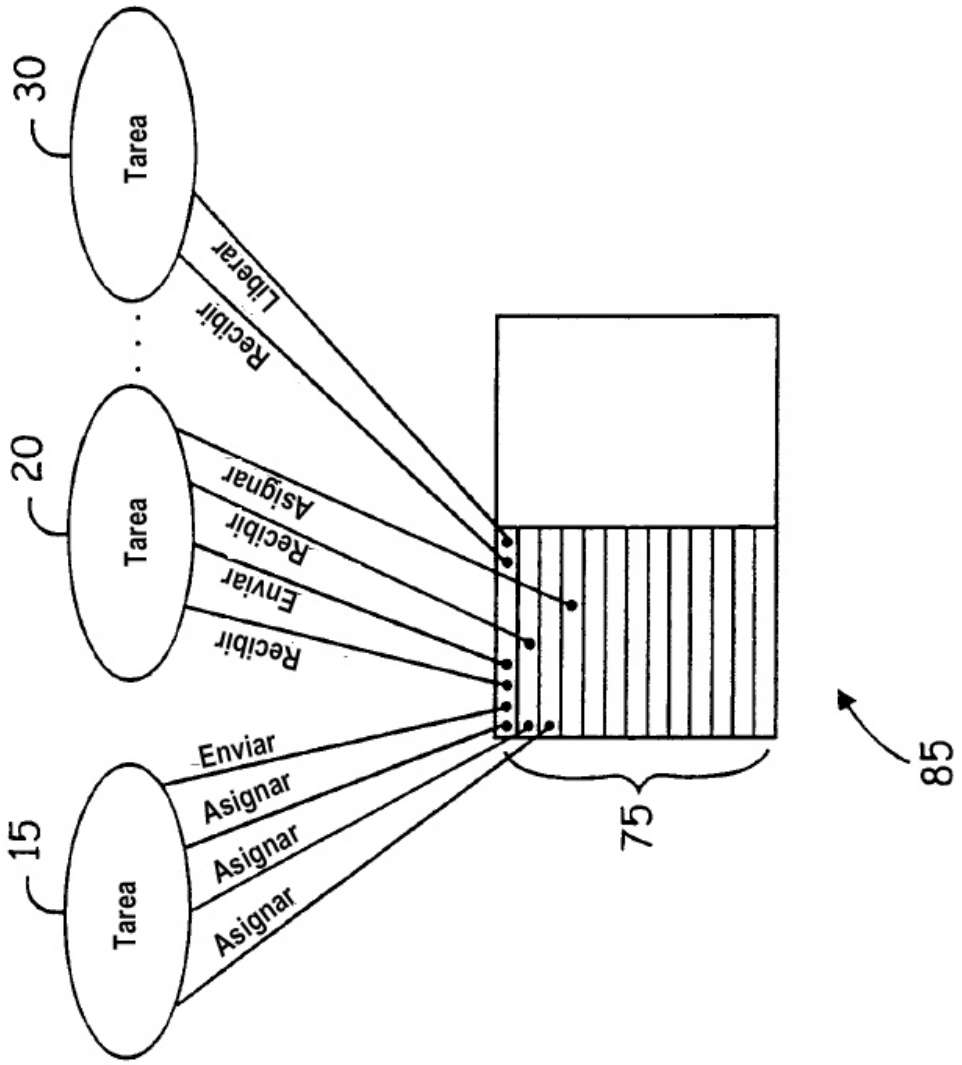
30
Tarea
Liberar
Recibir
20
Tarea
Asignar
Recibir
Enviar
Recibir
15
Tarea
Enviar
Asignar
Asignar
Asignar
75
85

Fig. 2C

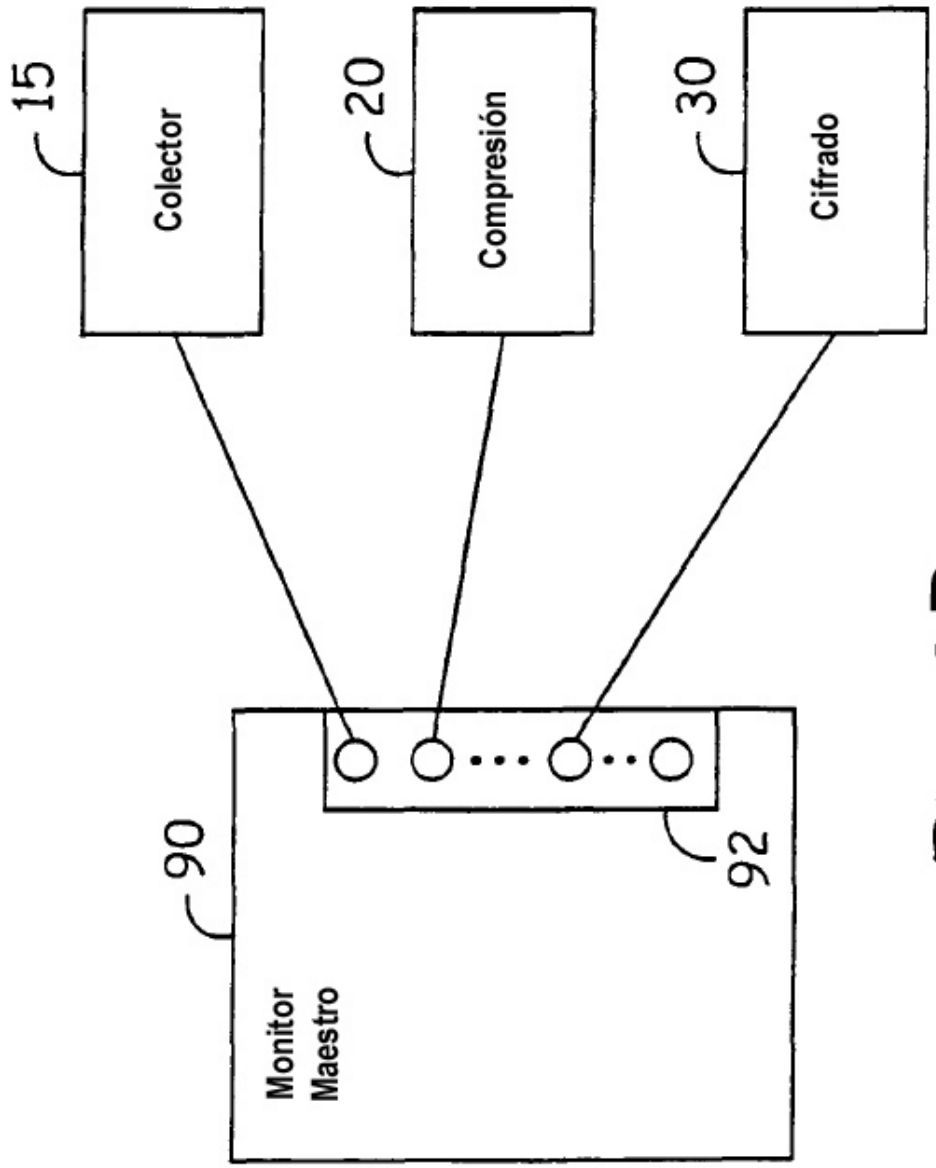

Fig. 2D

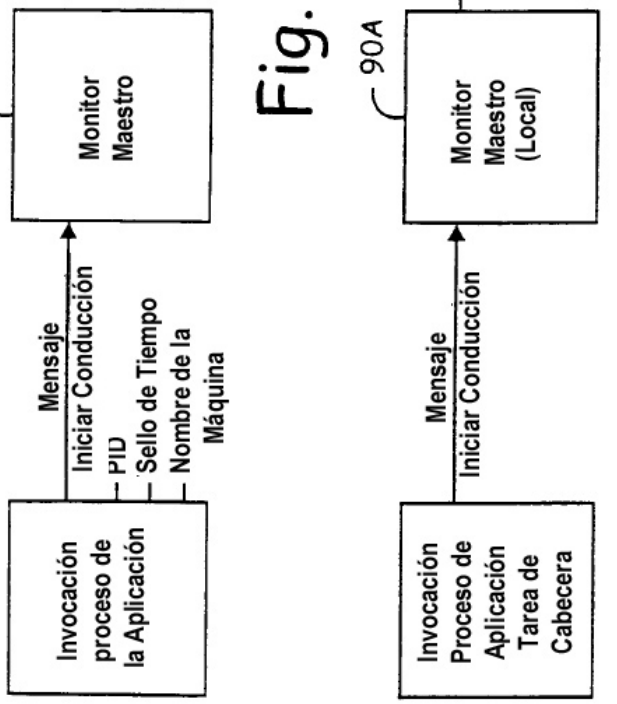

Fig. 3A

Fig. 3B

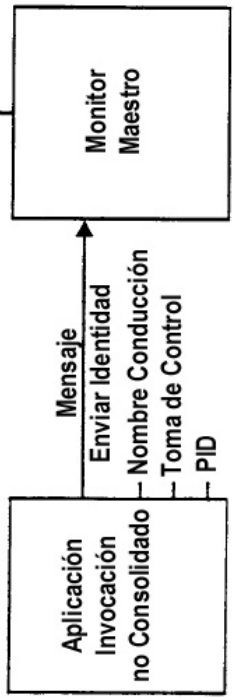

Fig. 3C

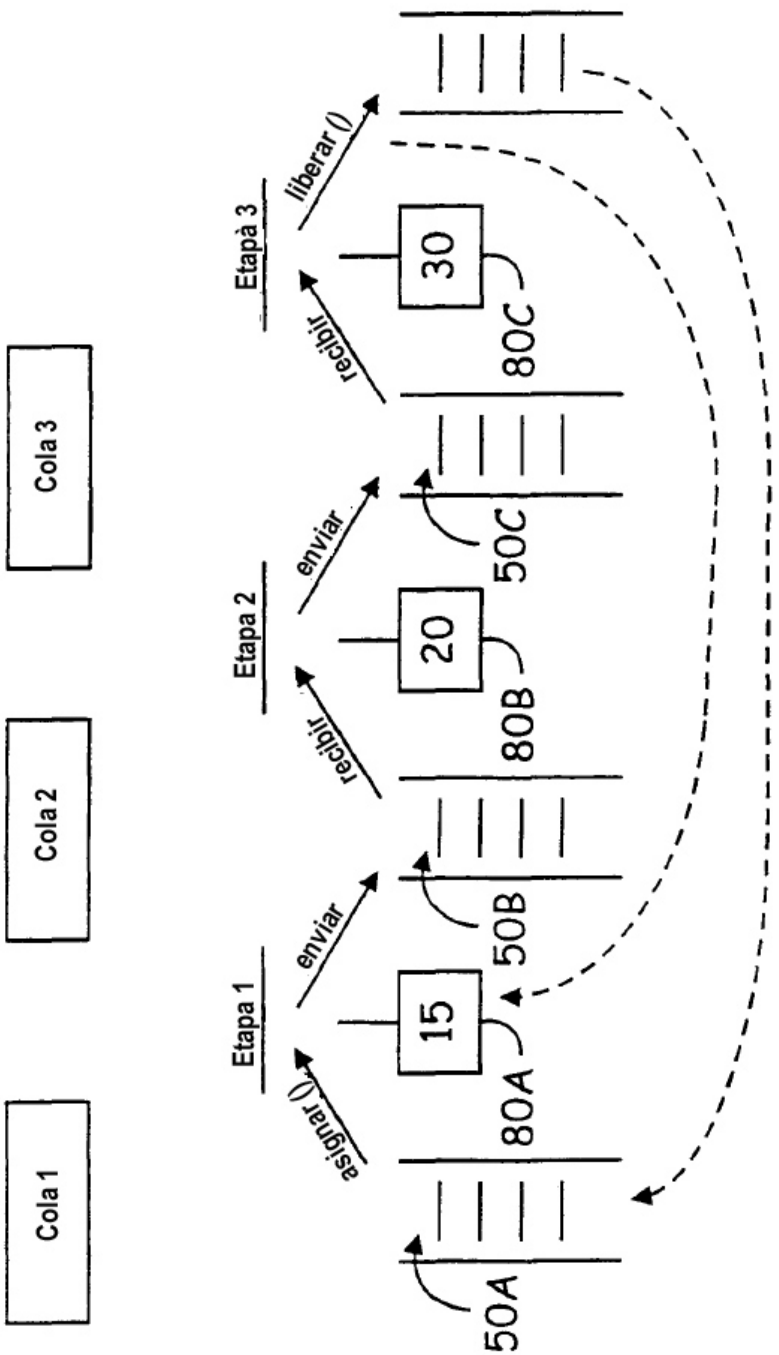
Cola 1
Cola 2
Cola 3
Etapa 1
Etapa 2
Etapa 3
asignar()
enviar
recibir
enviar
recibir
liberar()
15
20
30
50A
80A
50B
80B
50C
80C

Fig. 4A

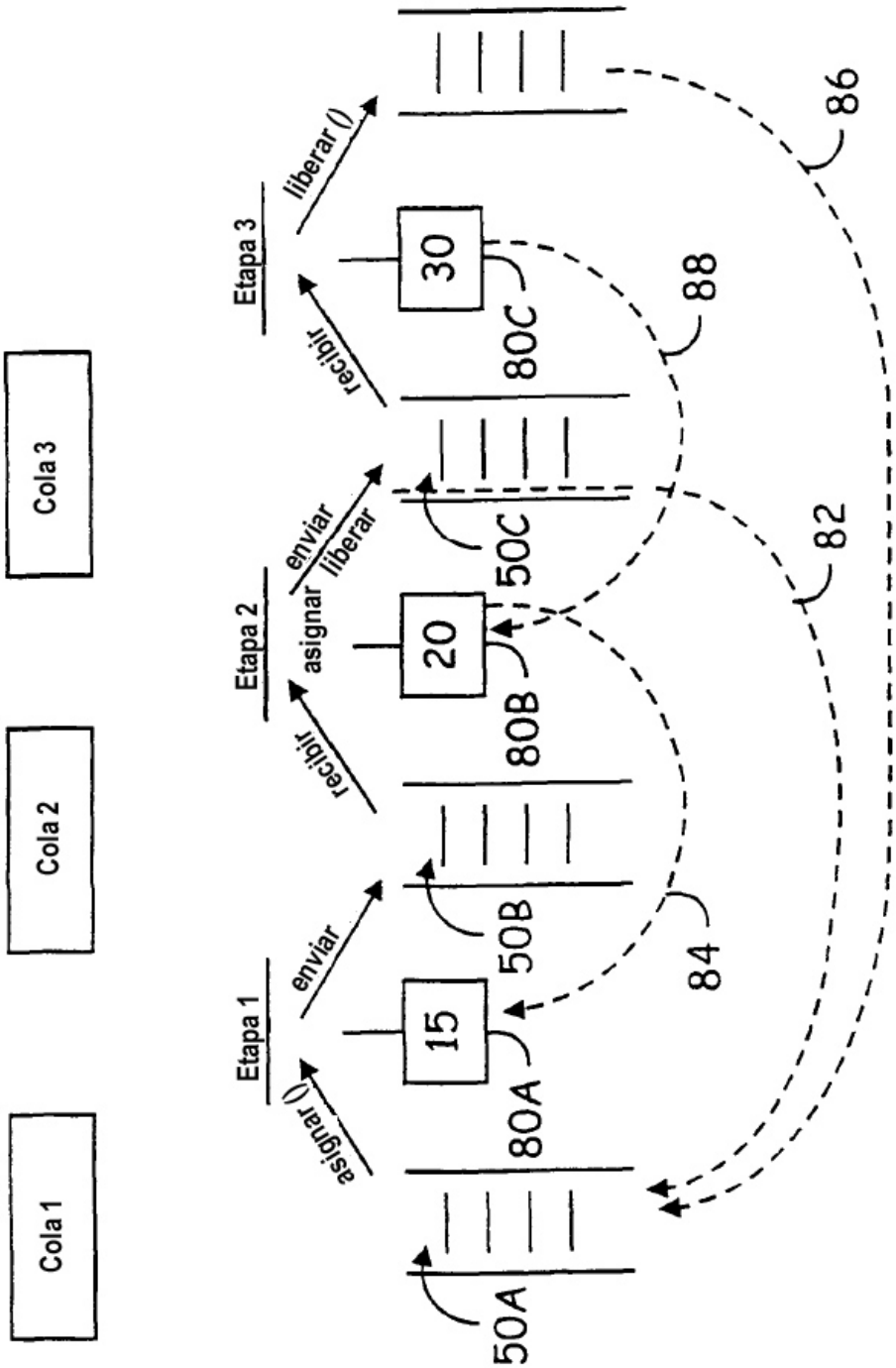
Cola 1
Cola 2
Cola 3
Etapa 1
Etapa 2
Etapa 3
asignar ()
enviar
recibir
asignar
enviar
liberar
recibir
liberar ()
15
20
30
50A
50B
50C
80A
80B
80C
82
84
86
88

Fig. 4B

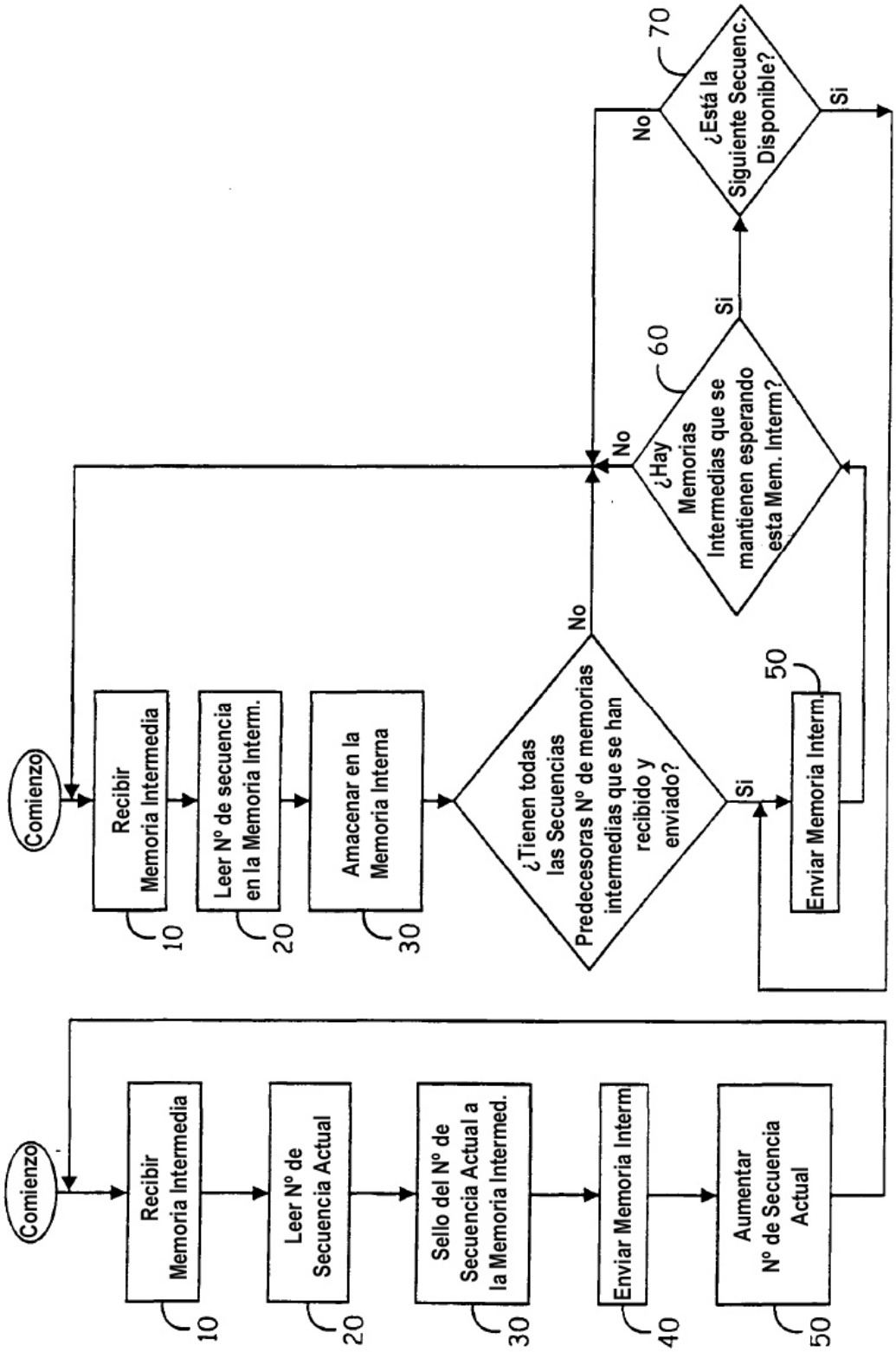
Comienzo
Recibir Memoria Intermedia
10
Leer Nº de Secuencia Actual
20
Sello del Nº de Secuencia Actual a la Memoria Intermed.
30
Enviar Memoria Interm
40
Aumentar Nº de Secuencia Actual
50

Fig. 5A

Comienzo
Recibir Memoria Intermedia
10
Leer Nº de secuencia en la Memoria Interm.
20
Amacenar en la Memoria Interna
30
¿Tienen todas las Secuencias Predecesoras Nº de memorias intermedias que se han recibido y enviado?
No
Si
Enviar Memoria Interm.
50
¿Hay Memorias Intermedias que se mantienen esperando esta Mem. Interm?
60
No
Si
¿Está la Siguiente Secuenc. Disponible?
70
No
Si

Fig. 5B

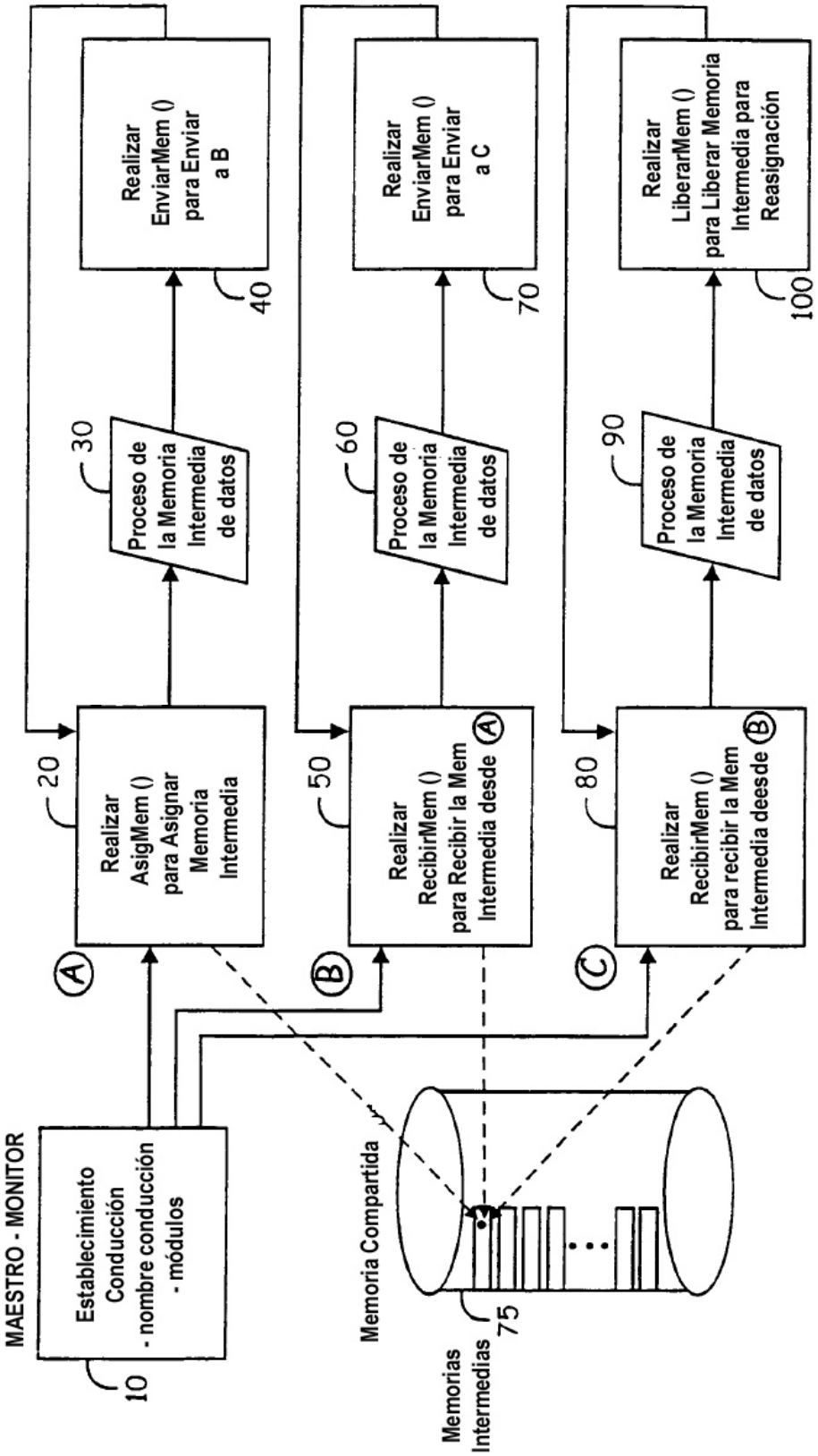
MAESTRO - MONITOR
Establecimiento Conducción - nombre conducción - módulos
10
Realizar AsigMem () para Asignar Memoria Intermedia
20
Realizar RecibirMem () para Recibir la Mem Intermedia desde A
50
Realizar RecibirMem () para recibir la Mem Intermedia deesde B
80
Proceso de la Memoria Intermedia de datos
30
Proceso de la Memoria Intermedia de datos
60
Proceso de la Memoria Intermedia de datos
90
Realizar EnviarMem () para Enviar a B
40
Realizar EnviarMem () para Enviar a C
70
Realizar LiberarMem () para Liberar Memoria Intermedia para Reasignación
100
A
B
C
Memoria Compartida
Memorias Intermedias
75

Fig. 6

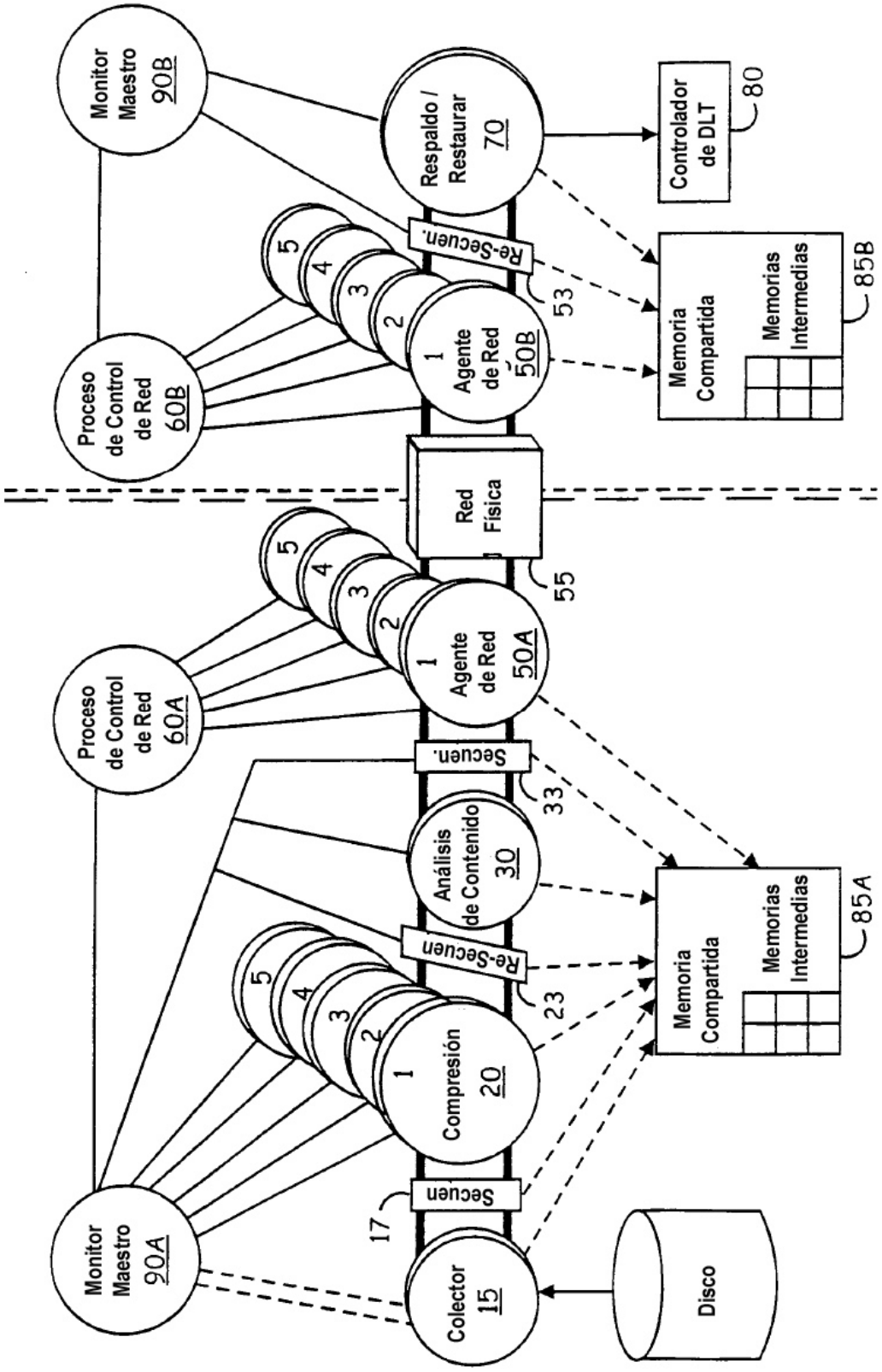
Monitor Maestro 90B
Respaldo / Restaurar 70
Controlador de DLT
80
Re-Secuen.
53
Proceso de Control de Red 60B
Agente de Red 50B
Memoria Compartida
Memorias Intermedias
85B
Red Física
55
Proceso de Control de Red 60A
Agente de Red 50A
Secuen.
33
Análisis de Contenido 30
Re-Secuen
23
Compresión 20
Memoria Compartida
Memorias Intermedias
85A
Secuen
17
Monitor Maestro 90A
Colector 15
Disco

Fig. 7

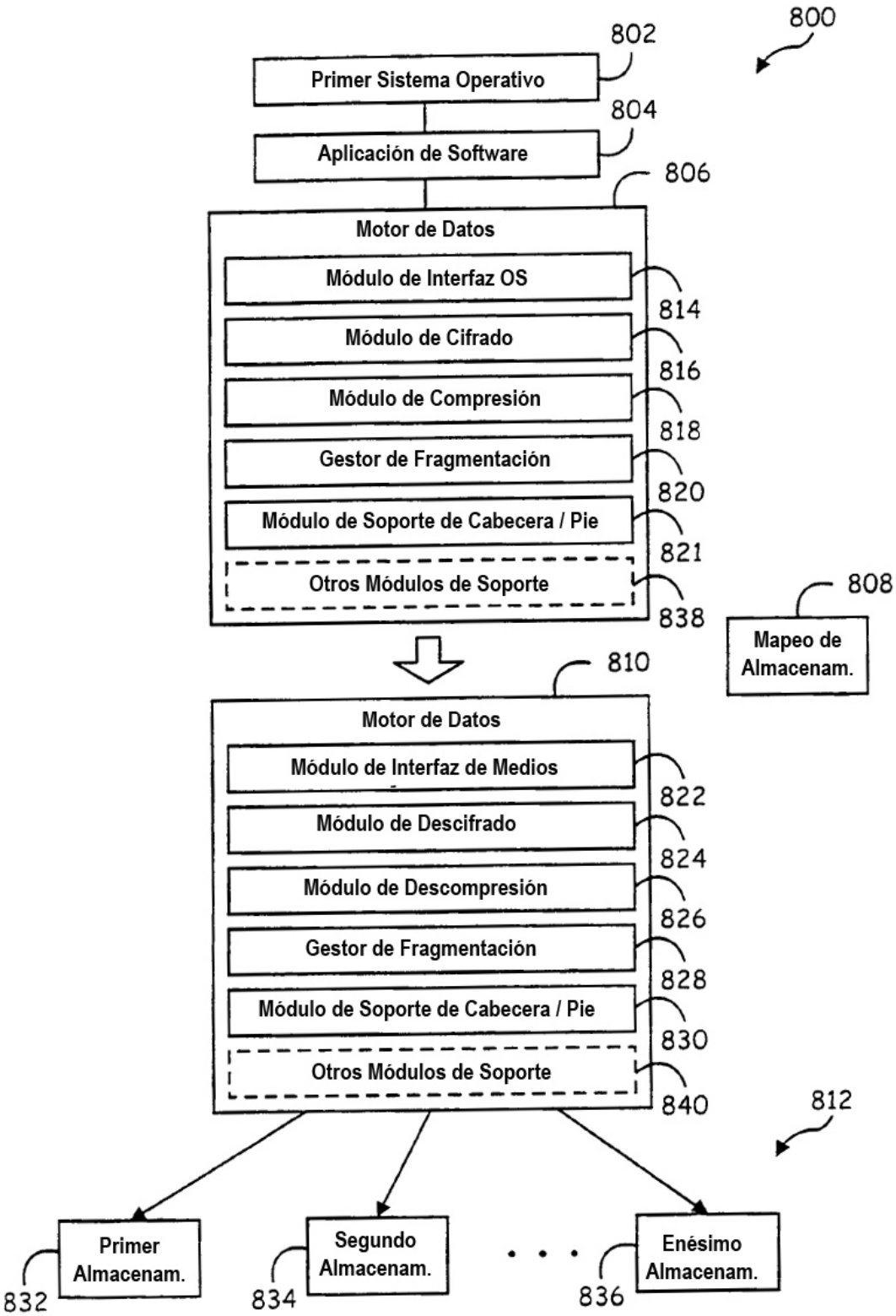
800
802
Primer Sistema Operativo
804
Aplicación de Software
806
Motor de Datos
Módulo de Interfaz OS
814
Módulo de Cifrado
816
Módulo de Compresión
818
Gestor de Fragmentación
820
Módulo de Soporte de Cabecera / Pie
821
Otros Módulos de Soporte
838
808
Mapeo de Almacenam.
810
Motor de Datos
Módulo de Interfaz de Medios
822
Módulo de Descifrado
824
Módulo de Descompresión
826
Gestor de Fragmentación
828
Módulo de Soporte de Cabecera / Pie
830
Otros Módulos de Soporte
840
812
Primer Almacenam.
832
Segundo Almacenam.
834
Enésimo Almacenam.
836

Fig. 8

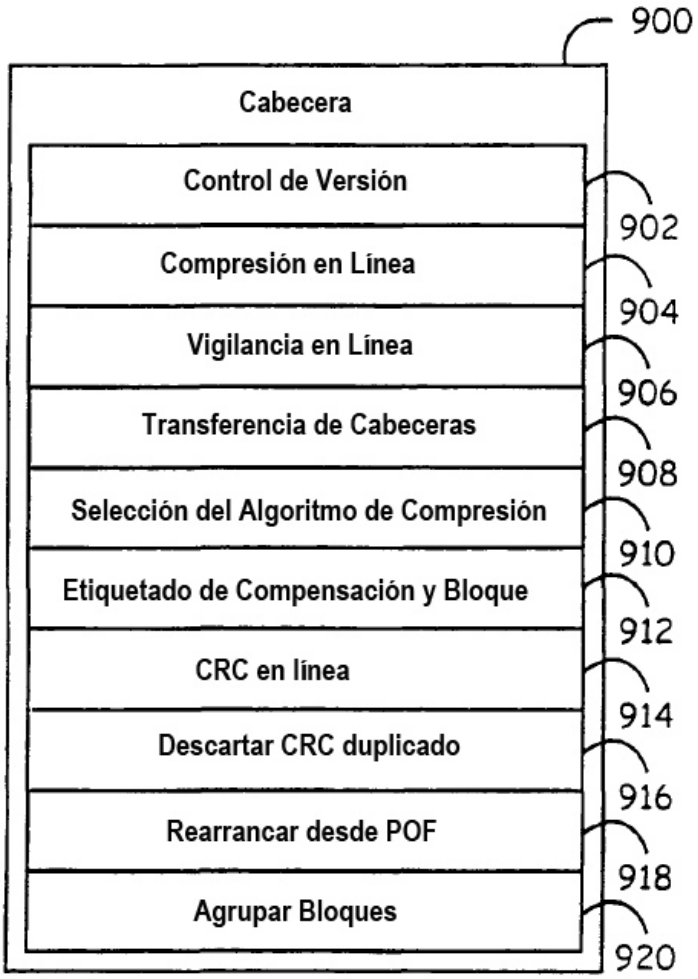
900
Cabecera
Control de Versión
902
Compresión en Línea
904
Vigilancia en Línea
906
Transferencia de Cabeceras
908
Selección del Algoritmo de Compresión
910
Etiquetado de Compensación y Bloque
912
CRC en línea
914
Descartar CRC duplicado
916
Rearrancar desde POF
918
Agrupar Bloques
920

Fig. 9

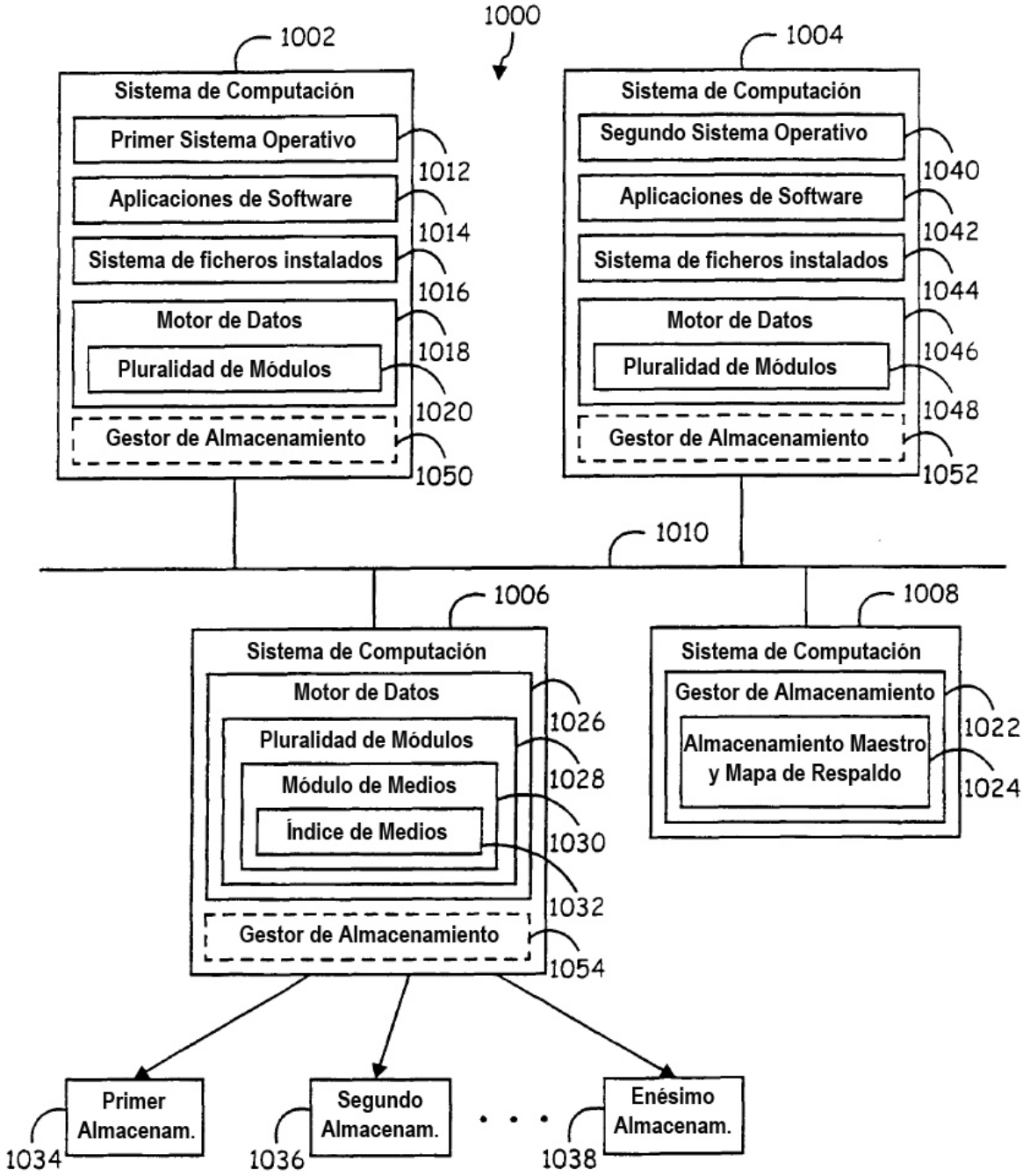
1000
1002
Sistema de Computación
Primer Sistema Operativo
1012
Aplicaciones de Software
1014
Sistema de ficheros instalados
1016
Motor de Datos
Pluralidad de Módulos
1018
1020
Gestor de Almacenamiento
1050
1004
Sistema de Computación
Segundo Sistema Operativo
1040
Aplicaciones de Software
1042
Sistema de ficheros instalados
1044
Motor de Datos
Pluralidad de Módulos
1046
1048
Gestor de Almacenamiento
1052
1010
1006
Sistema de Computación
Motor de Datos
1026
Pluralidad de Módulos
1028
Módulo de Medios
Índice de Medios
1030
1032
Gestor de Almacenamiento
1054
1008
Sistema de Computación
Gestor de Almacenamiento
1022
Almacenamiento Maestro y Mapa de Respaldo
1024
Primer Almacenam.
1034
Segundo Almacenam.
1036
Enésimo Almacenam.
1038

Fig. 10

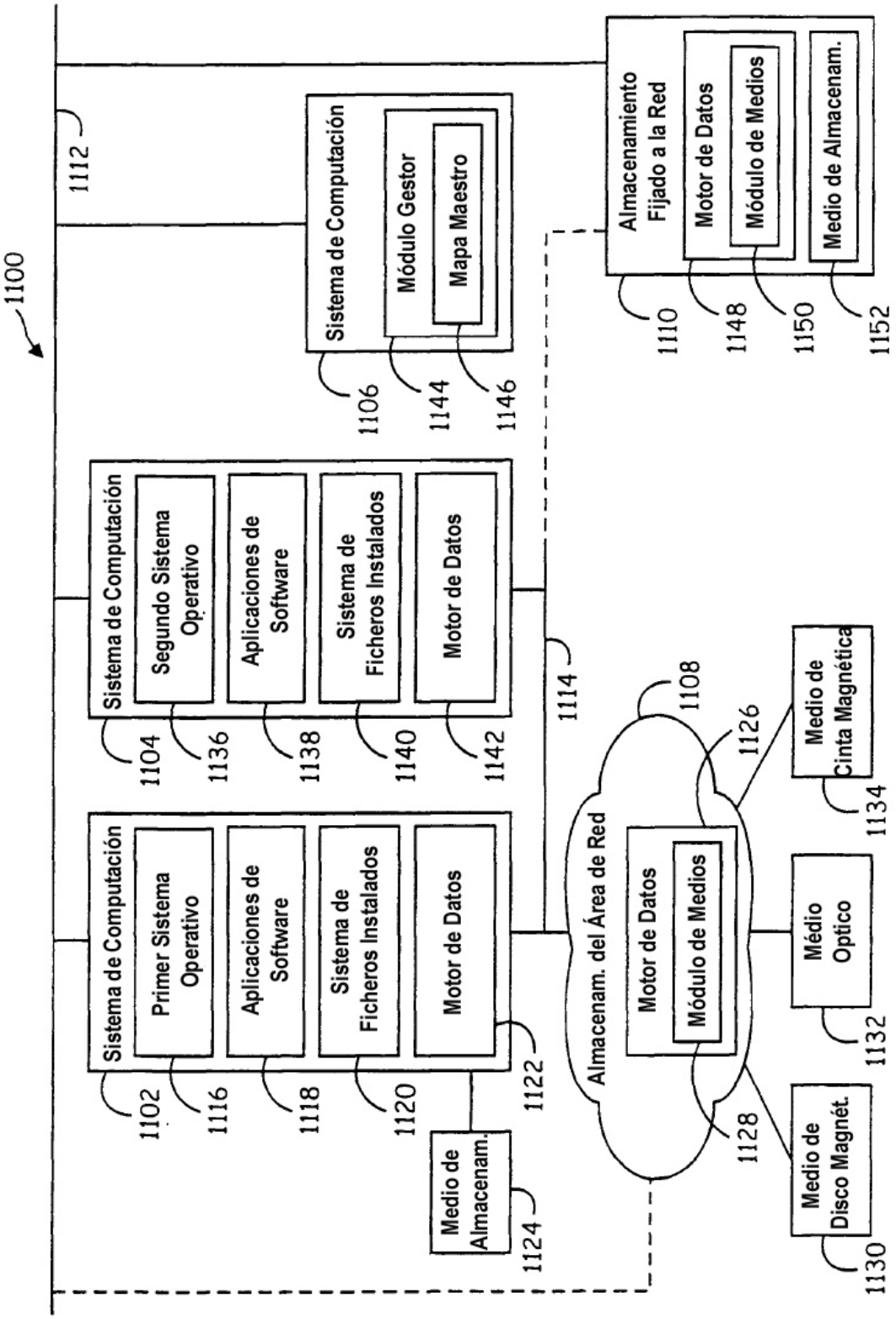
1100
1112
Sistema de Computación
Módulo Gestor
Mapa Maestro
1106
1144
1146
Almacenamiento Fijado a la Red
Motor de Datos
Módulo de Medios
Medio de Almacenam.
1110
1148
1150
1152
Sistema de Computación
Segundo Sistema Operativo
Aplicaciones de Software
Sistema de Ficheros Instalados
Motor de Datos
1104
1136
1138
1140
1142
1114
Sistema de Computación
Primer Sistema Operativo
Aplicaciones de Software
Sistema de Ficheros Instalados
Motor de Datos
1102
1116
1118
1120
1122
Medio de Almacenam.
1124
Almacenam. del Área de Red
Motor de Datos
Módulo de Medios
1108
1126
1128
Medio de Cinta Magnética
1134
Médio Optico
1132
Medio de Disco Magnét.
1130

Fig. 11

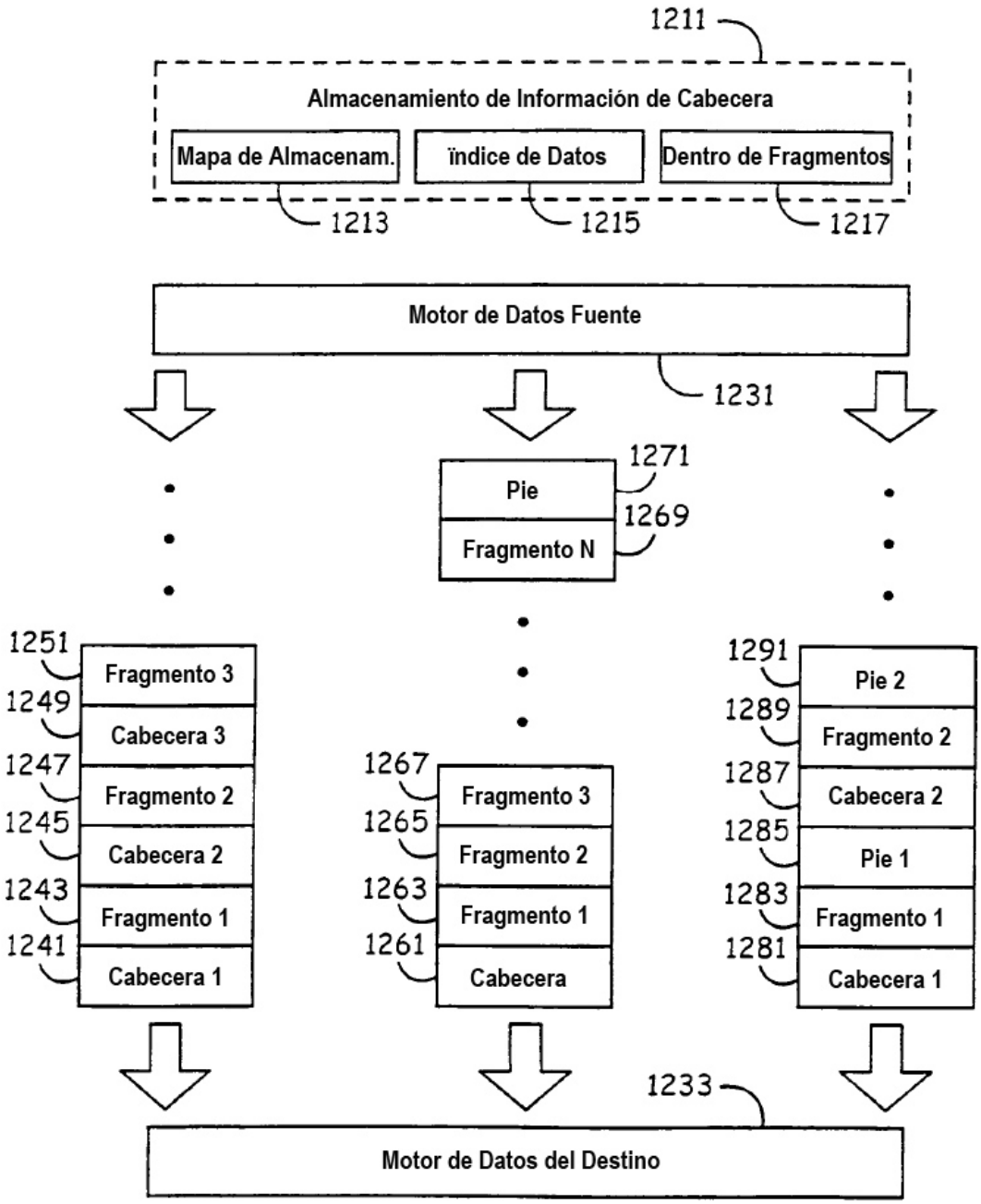
1211
Almacenamiento de Información de Cabecera
Mapa de Almacenam.
índice de Datos
Dentro de Fragmentos
1213
1215
1217
Motor de Datos Fuente
1231
1271
Pie
1269
Fragmento N
1251
Fragmento 3
1249
Cabecera 3
1247
Fragmento 2
1245
Cabecera 2
1243
Fragmento 1
1241
Cabecera 1
1267
Fragmento 3
1265
Fragmento 2
1263
Fragmento 1
1261
Cabecera
1291
Pie 2
1289
Fragmento 2
1287
Cabecera 2
1285
Pie 1
1283
Fragmento 1
1281
Cabecera 1
1233
Motor de Datos del Destino

Fig. 12

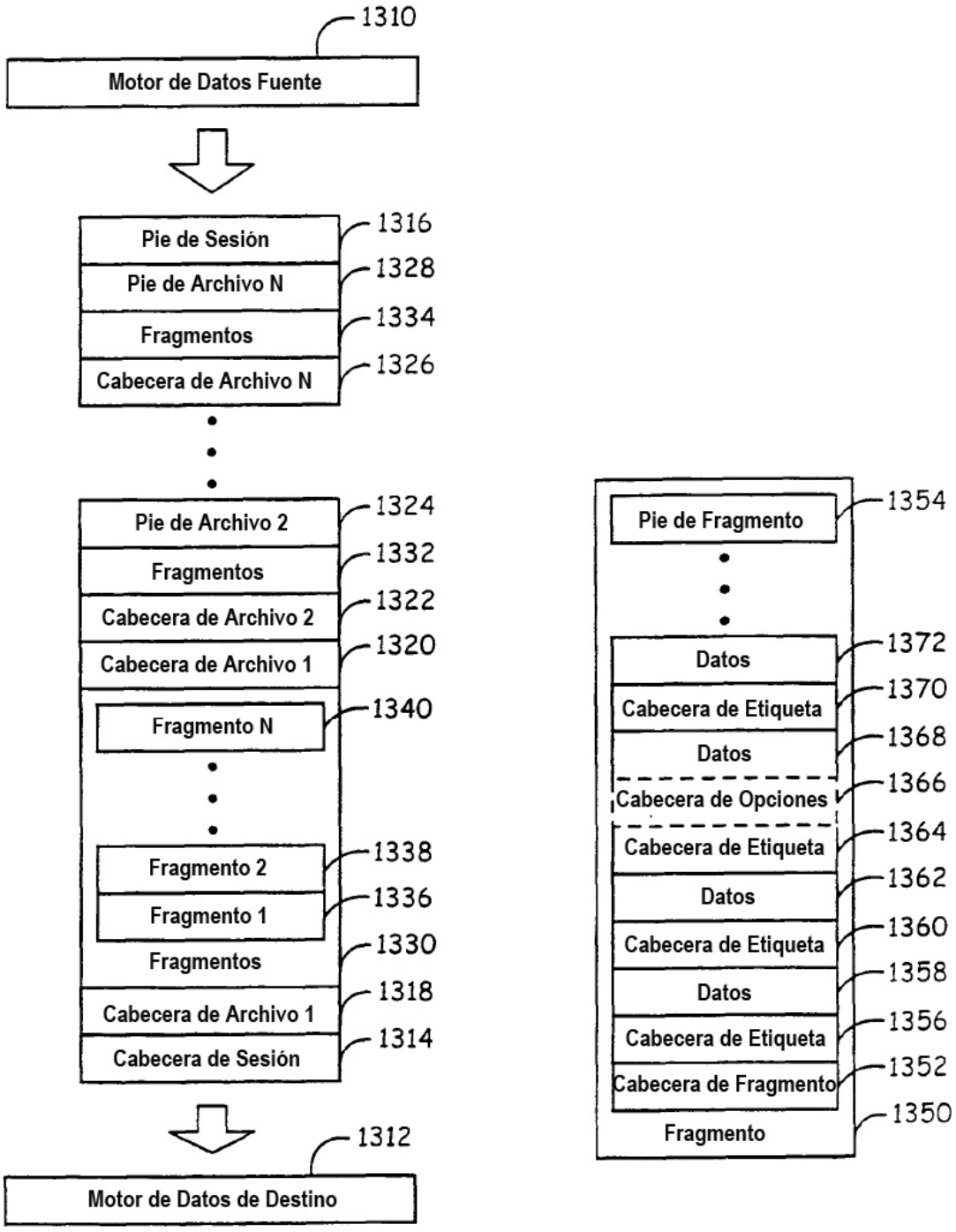
1310
Motor de Datos Fuente
Pie de Sesión
1316
Pie de Archivo N
1328
Fragmentos
1334
Cabecera de Archivo N
1326
Pie de Archivo 2
1324
Fragmentos
1332
Cabecera de Archivo 2
1322
Cabecera de Archivo 1
1320
Fragmento N
1340
Fragmento 2
1338
Fragmento 1
1336
Fragmentos
1330
Cabecera de Archivo 1
1318
Cabecera de Sesión
1314
1312
Motor de Datos de Destino
Pie de Fragmento
1354
Datos
1372
Cabecera de Etiqueta
1370
Datos
1368
Cabecera de Opciones
1366
Cabecera de Etiqueta
1364
Datos
1362
Cabecera de Etiqueta
1360
Datos
1358
Cabecera de Etiqueta
1356
Cabecera de Fragmento
1352
Fragmento
1350

Fig. 13