(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 645 486**

(51) Int. CI.:

G06F 13/42 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **22.07.2014 PCT/CN2014/082724**

(87) Fecha y número de publicación internacional: **29.01.2015 WO15010603**

(96) Fecha de presentación y número de la solicitud europea: **22.07.2014 E 14828761 (8)**

(97) Fecha y número de publicación de la concesión europea: **13.09.2017 EP 3025241**

(54) Título: **Comunicación internodal directa escalable sobre una interconexión de componentes periféricos expreso - (Peripheral Component Interconnect Express (PCIE))**

(30) Prioridad:

22.07.2013 US 201361857036 P
25.11.2013 US 201314089377

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
05.12.2017

(73) Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building, Bantian Longgang District , Shenzhen, Guangdong 518129, CN

(72) Inventor/es:

EGI, NORBERT y
SHI, GUANGYU

(74) Agente/Representante:

LEHMANN NOVO, María Isabel

DESCRIPCIÓN

Comunicación internodal directa escalable sobre una interconexión de componentes periféricos expreso – (Peripheral Component Interconnect Express (PCIE)).

Antecedentes

Los centros de datos pueden comprender grandes grupos de servidores. Los servidores de los centros de datos pueden aceptar solicitudes de los usuarios y responder a dichas solicitudes. Por ejemplo, los servidores pueden alojar datos y transmitir dichos datos a un usuario bajo solicitud. También se puede configurar un servidor para alojar procesos. Como tal, un usuario puede transmitir una solicitud a un servidor para realizar un proceso, el servidor puede realizar el proceso, y a continuación el servidor puede responder al usuario con los resultados del proceso. Un servidor puede comprender varios componentes para procesar las solicitudes de usuario y comunicarse con el usuario. Dichos servidores se pueden interconectar utilizando diferentes técnicas y dispositivos de red. Por ejemplo, un servidor se puede colocar en un chasis y se puede interconectar con otros servidores en otro chasis utilizando la capa dos del modelo de interconexión de sistemas abiertos (Open Systems Interconection (OSI)) (por ejemplo, el control de acceso a medios (Media Access Control (MAC)) y/o técnicas de la capa tres (por ejemplo, el protocolo internet (Internet Protocol (IP)).

El documento US 2009/248941A1 describe un método y una arquitectura de procesador de propósito especial punto a punto, que incluye un sistema con múltiples procesadores de propósito especial acoplados a, al menos, una unidad de procesamiento central por medio de un bus de puenteado anfitrión. Un bus directo acopla directamente cada uno de los múltiples procesadores de propósito especial a, al menos, otro de los múltiples procesadores de propósito especial. Un controlador de memoria se acopla a los múltiples procesadores de propósito especial y el controlador de memoria determina si transmitir los datos a través del bus anfitrión o el bus directo y si recibir los datos a través del bus anfitrión o el bus directo.

Resumen

En una forma de realización, la descripción incluye un método de comunicación de datos sobre un puenteado no transparente (Non-Transparent Bridge (NTB)) de la interconexión de componentes periféricos expreso (PCIe) que comprende transmitir un primer mensaje “posted write” a un procesador remoto a través del NTB, en donde el primer mensaje “posted write” indica una intención de transferir datos al procesador remoto y recibir un segundo mensaje “posted write” en respuesta al primer mensaje “posted write”, en donde el segundo mensaje “posted write” indica una lista de direcciones de destino para los datos.

En otra forma de realización, la descripción incluye un método de comunicación de datos sobre un NTB PCIe que comprende transmitir un primer mensaje “posted write” a un procesador remoto a través del NTB, en donde el primer mensaje “posted write” comprende una solicitud para leer datos y recibir un mensaje de transferencia de datos que comprende al menos algunos de los datos solicitados por el primer mensaje “posted write”.

En otra forma de realización, la descripción incluye un procesador, que comprende un módulo de transferencia de datos (234), que se configura para implementar una cola de recepción, una cola de transmisión y una cola de finalización;

y el procesador se configura para acoplarse a un NTB PCIe y leer datos de y escribir datos en varios procesadores remotos a través de la cola de recepción, la cola de transmisión, la cola de finalización y el NTB PCIe utilizando únicamente mensajes “write posted” y sin utilizar mensajes “non-posted”.

Estas y otras características se comprenderán más claramente a partir de la siguiente descripción detallada tomada en conjunto con los dibujos y reivindicaciones adjuntas.

Breve descripción de los dibujos

Para una comprensión más completa de esta descripción, se hace ahora referencia a la siguiente breve descripción, tomada en relación con los dibujos adjuntos y la descripción detallada, en donde los números de referencia similares representan partes similares.

La FIG. 1 es un diagrama esquemático de una forma de realización de una arquitectura de red de centro de datos desagregados.

La FIG. 2 es un diagrama esquemático de una forma de realización de un elemento de red (Network element (NE)), que puede actuar como un nodo dentro de una arquitectura de red de centro de datos desagregados.

La FIG. 3 es un diagrama de protocolo de una forma de realización de un método de escritura de datos utilizando únicamente mensajes “write post”.

La FIG. 4 es un diagrama de flujo de una forma de realización de otro método de escritura de datos utilizando únicamente mensajes “write post”.

La FIG. 5 un diagrama de protocolo de una forma de realización de un método de lectura de datos utilizando únicamente mensajes “write post” cuando se conoce el tamaño de los datos.

La FIG. 6 un diagrama de protocolo de una forma de realización de un método de lectura de datos utilizando únicamente mensajes “write post” cuando se desconoce el tamaño de los datos.

La FIG. 7 es un diagrama de flujo de otra forma de realización de un método de lectura de datos utilizando únicamente mensajes “write post”.

Descripción detallada

En contraste con una arquitectura de centro de datos que comprende varios servidores autónomos, una arquitectura de centro de datos desagregados se puede emplear para dar soporte a conjuntos de módulos de recursos. Dichos módulos de recursos se pueden no colocar en un chasis común y se pueden interconectar de una manera que permita el intercambio de recursos dinámico. Dichos módulos también se pueden diseñar para la compatibilidad retroactiva de tal manera que las actualizaciones de los centros de datos se puedan acometer sobre una base de modulo por modulo con un buen nivel de detalle en lugar de sobre una base de servidor por servidor más cara. Por ejemplo, un centro de datos que comprenda recursos de procesado insuficientes se puede equipar con un único módulo de procesado adicional en lugar de actualizarlo con un servidor completo que comprenda procesadores, memoria, circuitos de aceleración de procesos dedicados, una tarjeta de interfaz de red (Network Interface Card (NIC)), etc. En una arquitectura desagregada, los módulos de recursos se pueden conectar a través de una interconexión unificada. Una interconexión unificada de este tipo se puede implementar utilizando interconexión de componentes periféricos expreso (PCIe). Los módulos de procesado conectados a través de PCIe pueden emplear cada uno un espacio de direcciones de memoria significativo localmente. Dichos módulos de procesado se pueden conectar a través de un puenteado no transparente (NTB) PCIe, que puede traducir las direcciones significativas localmente a direcciones entendidas por la totalidad de la red y/o a un espacio de direcciones empleado por un módulo de procesado remoto. Cada procesador se puede asociar con un identificador solicitante (ID).

Los sistemas PCIe pueden emplear mensajes “posted” y mensajes “non-posted”. Un mensaje “posted” puede ser un mensaje que se puede tratar mediante hardware asociado ya que no requiere una respuesta. Los mensajes de escritura en memoria pueden ser mensajes “posted”. Un mensaje “non-posted” puede ser un mensaje que se puede tratar mediante hardware asociado ya que se requiere una respuesta. Los mensajes de lectura de memoria, los mensajes de lectura y/o escritura de entrada salida (E/S) y/o los mensajes de lectura y/o escritura de configuración pueden ser mensajes “non-posted”. Un NTB puede enrutar mensajes almacenando con estado un ID solicitante asociado con cada mensaje “non-posted” en una tabla de búsqueda de ID solicitantes (requestor ID Look-Up Table (R-LUT)). Después de recibir una respuesta a un mensaje de solicitud “non-posted”, el NTB puede emplear la R-LUT para determinar con qué mensaje de solicitud está asociada la respuesta y determinar dónde enviar la respuesta. Las tablas LUT-R NTB se pueden no diseñar para soportar la interconexión a gran escala. Por ejemplo, una R-LUT NTB puede comprender espacio de memoria insuficiente para soportar más de treinta y dos procesadores conectados simultáneamente. Como tal, la R-LUT NTB PCIe puede crear dificultades en la implementación de conjuntos de recursos de procesamiento a gran escala en una arquitectura de centro de datos desagregados.

En la presente memoria se describe un mecanismo para evitar la R-LUT NTB PCIe de una manera retrocompatible para permitir la creación de conjuntos de recursos de procesamiento a gran escala. Cada procesador se puede configurar para comunicarse exclusivamente con mensajes “posted” (por ejemplo, escrituras de memoria) ya que dichos mensajes pueden no utilizar todas las entradas disponibles en la R-LUT NTB PCIe. La gestión de dichos mensajes “posted” se puede llevar a cabo a nivel de software (por ejemplo, en lugar de a nivel de hardware) de manera que dichos mensajes “posted” pueden o no generar respuestas según sea necesario. Cada procesador se puede configurar para incluir una cola de recepción (RX), una cola de transmisión (TX) y una cola de finalización. Cada mensaje “posted” se puede analizar en base al contenido del mensaje y colocar en una cola asociada. El procesador puede actuar a continuación sobre cada mensaje en base a la cola a la que se ha asignado el mensaje. Por ejemplo, los mensajes que indican que el procesador debe prepararse para recibir una transferencia de datos se pueden colocar en la cola RX. Los mensajes que indican que el procesador debe prepararse para realizar una transferencia de datos se pueden colocar en la cola TX. Los mensajes que indican que ha sido finalizada la transferencia de datos se pueden colocar en la cola de finalización. Mediante el empleo de las colas RX, TX y de finalización, un procesador puede configurar y realizar transferencias de datos (por ejemplo, lecturas y escrituras de datos) con otros procesadores sobre un NTB utilizando únicamente mensajes “posted” (por ejemplo, mensajes de escritura) y puede evitar de este modo las limitaciones de escalabilidad asociadas con el NTB R-LUT. Aunque los mecanismos descritos en la presente memoria se pueden emplear para dar soporte a una arquitectura de centro de datos desagregados, se debe señalar que se pueden emplear dichos mecanismos para dar soporte a la conectividad basada en PCIe en cualquier otra arquitectura de centro de datos, como los centros de datos basados en servidores.

La FIG. 1 es un diagrama esquemático de una forma de realización de una arquitectura de red de centro de datos desagregados 100. La red 100 puede comprender un conjunto de módulos de procesado 110, un conjunto de módulos de memoria de proceso 150, un conjunto de módulos de almacenamiento de datos 120, un conjunto de módulos de aceleración de proceso 160 y un conjunto de módulos NIC 130 que se pueden conectar a través de una red de interconexión unificada 170. Los módulos de procesado 110, los módulos de memoria de proceso 150, los

módulos de almacenamiento de datos 120, los módulos de aceleración de proceso 160, los módulos NIC 130 y la red de interconexión unificada 170 se pueden colocar en un centro de datos común y se pueden no colocar en un recinto común (por ejemplo, cada módulo puede comprender un servidor separado, un servidor blade, un elemento de red, un chasis, etc.). Cada conjunto de módulos puede comprender varios módulos de recursos configurados cada uno para realizar una función común. Los módulos de procesado 110 pueden cada uno compartir el acceso a los recursos de los otros módulos a través de la red de interconexión unificada 170. La red de interconexión unificada 170 puede emplear un protocolo común a todos los módulos, tal como PCIe, lo que puede permitir que los módulos individuales se actualicen, añadan, y/o eliminen sin crear incompatibilidad de módulos. La capacidad de los módulos de procesado 110 para compartir recursos también puede permitir el equilibrio de la carga de recursos y puede reducir los cuellos de botella del proceso.

Cada módulo (por ejemplo, los módulos de procesado 110, los módulos de memoria de proceso 150, los módulos de almacenamiento de datos 120, los módulos de aceleración de proceso 160 y/o los módulos NIC 130) puede comprender y/o consistir esencialmente en los componentes necesarios para realizar una parte de una tarea y se puede colocar en un NE separado de todos los otros módulos. Por ejemplo, los módulos de procesado 110 pueden comprender y/o consistir esencialmente en un procesador 115, que puede ser un único procesador y/o un grupo de procesadores. El módulo de procesado 110 también puede opcionalmente comprender y/o consistir esencialmente en memoria de proceso local 117 y de almacenamiento local 113, así como componentes de transmisión para conectarse a la red de interconexión unificada 170 y los componentes relacionados con la alimentación. Los módulos de procesado 110 se pueden colocar en un servidor blade, que puede ser menos costoso y físicamente más pequeño que los servidores en bastidor y puede no poder proporcionar la funcionalidad completa sin acceso a la red de interconexión unificada 170. Los módulos de procesado 110 pueden operar para gestionar las tareas de los centros de datos típicas tales como gestionar el almacenamiento de datos, alojar los procesos, responder las consultas de clientes, etc.

La red 100 puede comprender un conjunto de módulos de memoria de proceso 150, que puede comprender y/o consistir esencialmente en memoria (por ejemplo, memoria de acceso aleatorio (Random Access Memory (RAM)), memoria caché del procesador, etc.) que pueden almacenar datos del procesador relacionados con los procesos activos. Los módulos de memoria de proceso 150 pueden comprender recursos de almacenamiento que se pueden asignar a un procesador particular 115, un módulo de procesado particular 110 y/o intercambiar por varios módulos de procesado 110. La asignación de módulos de memoria 150 se puede cambiar dinámicamente en base a las necesidades de la red 100 en un momento especificado. Un módulo de memoria de proceso 150 se puede colocar en un servidor blade. Por ejemplo, un módulo de memoria de proceso 150 puede consistir esencialmente en memoria, componentes de transmisión para dar soporte a la conexión con la red de interconexión unificada 170 y componentes de potencia.

La red 100 puede comprender un conjunto de módulos de almacenamiento de datos 120, que puede comprender y/o consistir esencialmente en dispositivos de almacenamiento de datos configurados para el almacenamiento a largo plazo (por ejemplo, unidades de disco, unidades de estado sólido, matriz redundante de discos independientes (Redundant Array of Independent Disk (RAID)), etc.) Los módulos de almacenamiento de datos 120 pueden comprender recursos de almacenamiento que se pueden asignar a un procesador particular 115, un módulo de procesado particular 110 y/o compartir por varios módulos de procesado 110. La asignación de módulos de almacenamiento de datos 120 se puede cambiar dinámicamente en base a las necesidades de la red 100 en un momento especificado. Un módulo de almacenamiento de datos 120 se puede colocar en un servidor blade. Por ejemplo, un módulo de almacenamiento de datos 120 puede consistir esencialmente en dispositivo(s) de almacenamiento de datos, componentes de transmisión para dar soporte a la conexión con la red de interconexión unificada 170 y componentes de potencia.

La red 100 puede comprender un conjunto de módulos de aceleración de proceso 160, que puede comprender y/o consistir esencialmente en aceleradores de proceso, tales como los circuitos integrados de aplicación específica (Application Specific Integrated Circuits (ASIC)) 163, las matrices de puertas programables en campo (Field Programmable Gate Array (FPGA)) 162, las unidades de procesamiento gráfico (Graphics Processing Units (GPU)) 161, los procesadores de señales digitales (Digital Signal Processor (DSP)), etc. Los aceleradores de proceso se pueden optimizar para una tarea específica y pueden realizar dichas tareas específicas de forma más rápida y/o eficiente que una unidad de procesamiento general (por ejemplo, los procesadores 115). Un procesador 115 puede desear descargar todo o parte de un proceso particular y puede transmitir una solicitud de recursos a los módulos de aceleración de proceso 160 y los módulos de aceleración de proceso 160 pueden emplear los aceleradores de proceso para completar el proceso y transmitir los datos resultantes de vuelta al procesador 115 solicitante. Los módulos de aceleración de proceso 160 pueden comprender recursos de procesamiento que se pueden asignar a un procesador particular 115, un módulo de procesado 110 particular y/o compartir por varios módulos de procesado 110. La asignación de un módulo de aceleración de proceso 160 se puede cambiar dinámicamente en base a las necesidades de la red 100 en un momento especificado. Un módulo de aceleración de proceso 160 se puede colocar en un servidor blade. Por ejemplo, un módulo de aceleración de proceso 160 puede consistir esencialmente en un acelerador de proceso (por ejemplo, el ASIC 163, la FPGA 162 y/o la GPU 161), componentes de transmisión para dar soporte a la conexión con la red de interconexión unificada 170 y componentes de potencia.

La red 100 puede comprender un conjunto de módulos NIC 130, que pueden comprender y/o consistir esencialmente en NIC configuradas para comunicarse con una red principal de centro de datos 140, el internet y/o un dispositivo cliente local 145 en nombre de los otros módulos. Como un ejemplo, los módulos NIC 130 pueden comprender recursos de conectividad que se pueden distribuir a un procesador particular 115, un módulo de procesado particular 110 y/o compartir por varios módulos de procesado 110. La asignación de un módulo NIC 130 y/o los recursos de módulos NIC 130 se pueden cambiar dinámicamente en base a las necesidades de la red 100 en un momento especificado. Como otro ejemplo, los módulos NIC 130 se pueden configurar para comunicarse con la red principal en nombre de los módulos de procesado 110, los módulos de aceleración de proceso 160, los módulos de memoria de proceso 150, los módulos de almacenamiento 120 o combinaciones de los mismos. Como tal, un módulo de procesado 110 puede dirigir otros módulos a comunicar la salida directamente al NIC 130 sin volver a un módulo de procesado 110. Un módulo NIC 130 se puede colocar en un servidor blade. Por ejemplo, un módulo NIC 130 puede consistir esencialmente en NIC para la comunicación con la red principal 140, componentes de transmisión para dar soporte a la conexión con la red de interconexión unificada 170 y componentes de potencia. Los módulos NIC también pueden implementar el acceso remoto directo a memoria (Remote Direct Memory Access (RDMA)).

Los conjuntos de módulos (por ejemplo, los módulos de procesado 110, los módulos de memoria de proceso 150, los módulos de almacenamiento de datos 120, los módulos de aceleración de proceso 160 y/o los módulos NIC 130) se pueden interconectar mediante una red de interconexión unificada 170. La red de interconexión unificada 170 puede transportar las comunicaciones entre los módulos y/o conjuntos de una manera sin bloqueos. La red de interconexión unificada 170 puede comprender cualquier hardware y/o protocolos que puedan ser compatibles con todos los módulos. Por ejemplo, la red de interconexión unificada 170 puede comprender una red PCI-e. La red de interconexión unificada 170 se puede no restringir a un módulo particular (por ejemplo, colocado dentro de un servidor blade) y/o chasis y se puede enrutar a través de un centro de datos. Los módulos que comprenden componentes que no dan soporte de forma nativa a conexiones a través de la red de interconexión unificada 170 pueden comprender procesadores y/u otros componentes de conexión para dar soporte a la interconectividad.

La red de interconexión unificada 170 puede, por ejemplo, comprender varios NTB 171 compatibles con PCIe. Un NTB 171 puede actuar como puerta de enlace para las comunicaciones que pasan entre un procesador particular 115 y/o el módulo de procesado 110 y la interconexión unificada 170. Aunque cada procesador 115 y/o el módulo de procesado 110 se pueden conectar a un NTB 171 lógico dedicado, múltiples NTB 171 se pueden o no colocar en un dispositivo físico único (no mostrado). Cada procesador 115 y/o módulo de procesado 110 puede comprender un espacio de dirección de memoria localmente significativo que puede no ser reconocido por otros procesadores 115, módulos de procesado 110 y/u otros dispositivos de red 100. Cada NTB 171 se puede configurar para realizar la traducción de las direcciones de red en nombre del procesador 115 y/o el módulo de procesado 110 para permitir la comunicación con otros procesadores y/o módulos. Por ejemplo, un primer NTB 171 conectado a un primer procesador 115 puede traducir los mensajes dirigidos en el espacio de direcciones del primer procesador 115 a un espacio de direcciones comprendido a través de la interconexión unificada 170 y viceversa. Del mismo modo, un segundo NTB 171 puede realizar las mismas traducciones para un segundo procesador 115 conectado, que puede permitir la comunicación entre el primer procesador 115 y el segundo procesador 115 a través de la traducción de direcciones en el primer NTB 171 y el segundo NTB 171.

Los procesadores 115 y/o módulos de procesado 110 pueden comunicarse a través de los NTB 171 a través de mensajes “posted” y mensajes “non-posted”. Un mensaje “posted” puede no requerir una respuesta, mientras que un mensaje “non-posted” puede requerir una respuesta. Un NTB 171 puede comprender una R-LUT. Cuando se recibe un mensaje “non-posted”, por ejemplo, desde un procesador remoto, un NTB 171 puede almacenar una ID de solicitante asociado con el procesador remoto en la R-LUT. Después de recibir una respuesta al mensaje “non-posted”, por ejemplo, desde un procesador local, la NTB 171 puede consultar la R-LUT para determinar dónde enviar la respuesta. Los NTB 171 R-LUT pueden ser con estados y se pueden diseñar para dar soporte a un número relativamente pequeño de procesadores (por ejemplo, un máximo de ocho o treinta y dos). Como tal, una NTB 171 R-LUT puede evitar la escalabilidad de la red 100 más allá de treinta y dos módulos de procesado 110. Sin embargo, los procesadores 115 se pueden configurar para evitar la R-LUT empleando solamente mensajes “posted”, lo que puede permitir la escalabilidad hasta aproximadamente sesenta y cuatro mil procesadores. Para gestionar las transacciones utilizando únicamente mensajes “posted”, pueden ser necesarios procesadores 115 y/o módulos de procesado 110 para gestionar las comunicaciones en el nivel de software en lugar de en el nivel de hardware. Por ejemplo, un procesador 115 se puede configurar con una cola RX, una cola TX y una cola de finalización. La(s) cola(s) RX, cola(s) TX y cola(s) de finalización se puede(n) configurar como colas “Primero en entrar, primero en salir (First In First Out (FIFO))”. Los procesadores 115 se pueden configurar para reconocer que un mensaje “posted write” puede no invocar una escritura y puede llevar, en cambio, otra información. Los procesadores 115 pueden analizar el contenido de un mensaje entrante (por ejemplo, paquete de datos) y colocar el mensaje en una cola de acuerdo con el contenido de los mensajes, por ejemplo, en base a la dirección y/o en base a un comando codificado en la carga útil del mensaje. Los mensajes relacionados con una transmisión inminente de datos se pueden colocar en la cola TX, los mensajes relacionados con una recepción inminente de datos se pueden colocar en la cola RX y los mensajes relacionados con la finalización de una transacción se pueden colocar en una cola de finalización. El procesador 115 y/o módulos de procesado 110 pueden a continuación tratar cada mensaje en base a la cola a la que se ha asignado el mensaje.

La FIG. 2 es un diagrama esquemático de una forma de realización de un NE 200, que puede actuar como un nodo (por ejemplo, un módulo de procesado 110) dentro de una arquitectura de red de centro de datos desagregados, tal como una arquitectura de red de centro de datos desagregados 100. Un experto en la técnica reconocerá que el término NE abarca un amplio rango de dispositivos de los que el NE 200 es meramente un ejemplo. El NE 200 se incluye para fines de claridad de la descripción, pero de ninguna manera se pretende limitar la aplicación de la presente descripción a una forma de realización particular del NE o clase de formas de realización del NE. Al menos algunas de las características/métodos descritos en la descripción se pueden implementar utilizando un aparato o componente de red tal como un NE 200. Por ejemplo, las características/métodos en la descripción se pueden implementar utilizando hardware, firmware y/o software instalado para ser ejecutado con el hardware. El NE 200 puede ser cualquier dispositivo que transporte tramas a través de una red, por ejemplo, un conmutador, un enrutador, un puenteado, un servidor, un cliente, etc. Según se muestra en la FIG. 2, el NE 200 puede comprender transceptores (Tx/Rx) 210, que pueden ser transmisores, receptores o combinaciones de los mismos. Un Tx/Rx 210 se puede acoplar a varios puertos aguas abajo 220 para transmitir y/o recibir tramas desde otros nodos, un Tx/Rx 210 acoplado a varios puertos aguas arriba 250 para transmitir y/o recibir tramas desde otros nodos. Un procesador 230 se puede acoplar a los Tx/Rx 210 para procesar las tramas y/o determinar a qué nodos enviar tramas. El procesador 230 puede comprender uno o más procesadores multinúcleo y/o dispositivos de memoria 232, que pueden funcionar como almacenes de datos, búferes, etc. El procesador 230 se puede implementar como un procesador general o puede ser parte de uno o más ASIC y/o DSP. El procesador 230 puede comprender un módulo de transferencia de datos 234, que puede implementar una cola RX, una cola TX, una cola de finalización y/o puede implementar leer y/o escribir operaciones utilizando únicamente mensajes post para evitar una R-LUT NTB PCIe. En una forma de realización alternativa, el módulo de transferencia de datos 234 se puede implementar como instrucciones almacenadas en la memoria 232, que se pueden ejecutar con el procesador 230. En otra forma de realización alternativa, el módulo de transferencia de datos 234 se puede implementar en NE separados. Los puertos aguas abajo 220 y/o los puertos aguas arriba 250 pueden contener componentes de transmisión y/o recepción eléctricos y/u ópticos. El NE 200 puede o no ser un componente de enrutamiento que tome decisiones de enrutamiento.

Se entiende que mediante programación y/o carga de instrucciones ejecutables en el NE 200, se cambia al menos uno de, el procesador 230, el módulo de transferencia de datos 234, los puertos aguas abajo 220, los Tx/Rx 210, la memoria 232 y/o los puertos aguas arriba 250, transformando el NE 200 en parte en una máquina o aparato particular, por ejemplo, una arquitectura de reenvío múltinúcleo, que tiene la nueva funcionalidad mostrada por la presente descripción. Es fundamental para las técnicas de la ingeniería de software y la ingeniería eléctrica que la funcionalidad que se puede implementar mediante carga de software ejecutable en un ordenador, se pueda convertir en una implementación de hardware mediante reglas de diseño bien conocidas. Las decisiones entre implementar un concepto en software frente a hardware normalmente dependen de consideraciones de estabilidad de diseño y números de unidades a producir más que de cualesquiera problemas relacionados con la traducción desde el dominio del software al dominio del hardware. Generalmente, se puede preferir que un diseño que todavía está sujeto a cambios frecuentes se implemente con software, porque volver a hacer una implementación de hardware es más caro que volver a hacer un diseño de software. Generalmente, se puede preferir un diseño que sea estable que se producirá en grandes volúmenes para ser implementado en hardware, por ejemplo, en un ASIC, porque para producciones grandes ejecutar la implementación con hardware puede ser menos costoso que la implementación con software. A menudo, un diseño se puede desarrollar y probar en una forma con software y transformar posteriormente, mediante reglas de diseño bien conocidas, a una implementación con hardware equivalente en un circuito integrado específico de aplicación que predetermina las instrucciones del software. De la misma manera que una máquina controlada por un nuevo ASIC es una máquina o aparato particular, asimismo un ordenador que ha sido programado y/o cargado con instrucciones ejecutables se puede ver como una máquina o aparato particular.

La FIG. 3 es un diagrama de protocolo de una forma de realización de un método 300 de escritura de datos utilizando únicamente mensajes “write post”. Por ejemplo, el método 300 se puede implementar en un procesador (por ejemplo, el procesador 115) y/o en un módulo de procesado (por ejemplo, el módulo de procesado 110). Un procesador de este tipo, denominado en la presente memoria como un primer procesador, un procesador local y/o Procesador 1, puede desear escribir datos a otro procesador, denominado en la presente memoria como un segundo procesador, procesador remoto y/o Procesador 2, a través de un NTB PCIe, como el NTB 171. Aunque el Procesador 1 puede operar en la red 100, se debe señalar que el Procesador 1 también se puede colocar en cualquier otra red basada en PCIe. El Procesador 2 puede o no ser, en el fondo, similar al Procesador 1 y puede o no ser colocado en el mismo chasis que el Procesador 1. El Procesador 1 y el Procesador 2 se pueden tanto configurar con una cola RX, una cola TX y una cola de finalización.

El Procesador 1 puede ser consciente del tamaño de los datos a enviar al Procesador 2. En la etapa 301, el Procesador 1 puede transmitir un mensaje de escritura post (por ejemplo, un paquete de datos) al Procesador 2. El mensaje de escritura post de la etapa 301 puede comprender información relacionada con los datos a enviar y puede incluir el tamaño de los datos. Como el Procesador 1 puede desear que el Procesador 2 reciba los datos, el mensaje de escritura post de la etapa 301 se puede transmitir a la cola RX del Procesador 2, por ejemplo, en base a una dirección asociada con la cola o basada en un comando codificado en la carga útil del mensaje. Una vez que el mensaje de la etapa 301 llega a la parte delantera de la cola RX, el Procesador 2 puede realizar la etapa 303 asignando memoria para recibir los datos en base al tamaño de los datos. El Procesador 2 también puede fijar

páginas virtuales asociadas para evitar que dichas páginas y datos asociados se intercambien (por ejemplo, se eliminen de la memoria a un disco duro) antes de finalizar la escritura indicada en la etapa 301. En la etapa 305, el Procesador 2 puede crear una lista de direcciones de destino, tal como una Lista de dispersión asociación (Scatter Gather List (SGL)), que comprende las direcciones de las ubicaciones de memoria asignadas para recibir los datos transmitidos. En la etapa 307, el Procesador 2 puede transmitir un mensaje de escritura post al Procesador 1. El mensaje de escritura post de la etapa 307 puede comprender la lista de direcciones de memoria de destino (por ejemplo, según se genera en la etapa 305). Como el mensaje de escritura post de la etapa 307 se puede referir a una transmisión de datos desde el Procesador 1, el mensaje de escritura post se puede transmitir a la cola TX del Procesador 1. Una vez que el mensaje de la etapa 307 llega a la parte delantera de la cola TX, el Procesador 1 puede realizar la etapa 309 moviendo los datos a las direcciones de memoria enumeradas en la lista de direcciones de destino. La Etapa 307 se puede realizar transmitiendo mensaje(s) de escritura post que comprenden los datos, empleando Acceso Directo a Memoria (Direct Memory Access (DMA)), etc. En la etapa 311, el Procesador 1 puede transmitir un mensaje de escritura post al Procesador 2 indicando que la transferencia de datos asociada ha sido finalizada. Como el mensaje de escritura post de la etapa 311 se refiere a un mensaje de finalización, el mensaje de escritura post de la etapa 311 se puede transmitir a la cola de finalización del Procesador 2. Después de recibir todos los datos, el Procesador 2 también puede transmitir un mensaje de finalización de escritura post al Procesador 1 en la etapa 313. El mensaje de la etapa 313 puede indicar que todos los datos han sido recibidos por el Procesador 2. Como el mensaje de escritura post de la etapa 313 se refiere a un el mensaje de finalización, el mensaje de escritura post de la etapa 313 se puede transmitir a la cola de finalización del Procesador 1. La etapa 313 puede ser opcional. La etapa 313 se ilustra como una flecha discontinua en la FIG. 3 para indicar la naturaleza opcional de la etapa 313.

La FIG.4 es un diagrama de flujo de una forma de realización de otro método 400 de escritura de datos utilizando únicamente mensajes “write post”. El método 400 se puede implementar con un procesador local (por ejemplo, un Procesador 1) que desee escribir datos en un procesador remoto (por ejemplo, el Procesador 2), ambos de los cuales pueden ser, en el fondo, similares a los procesadores descritos con referencia al método 300. En la etapa 401, un mensaje de escritura post se puede transmitir a una cola de recepción en un procesador remoto (por ejemplo, el Procesador 2). El mensaje de la etapa 401 puede indicar una intención de mover datos junto con el tamaño de los datos a transferir. En la etapa 403, se puede recibir un mensaje de escritura post del procesador remoto. El mensaje de escritura post de la etapa 403 puede comprender una SGL de direcciones de destino y se puede colocar en una cola de transmisión. En la etapa 405, se pueden emplear mensaje(s) de escritura post y/o DMA para transmitir los datos a las ubicaciones de memoria remotas indicadas en la SGL. En la etapa 407, un mensaje de escritura post se puede transmitir a una cola de finalización en el procesador remoto. El mensaje de la etapa 407 puede indicar que la transferencia de datos ha finalizado. En la etapa 409, se puede recibir un mensaje de escritura post en una cola de finalización. El mensaje de escritura post de la etapa 409 puede indicar que los datos han sido completamente recibidos en las ubicaciones de memoria remotas especificadas por la SGL recibida en la etapa 403.

La FIG. 5 es un diagrama de protocolo de una forma de realización de un método 500 de lectura de datos utilizando únicamente mensajes “write post” cuando se conoce el tamaño de datos. El método 500 se puede implementar con un procesador local (por ejemplo, un Procesador 1) que desee leer datos de un procesador remoto (por ejemplo, el Procesador 2), ambos de los cuales pueden ser, en el fondo, similares a los procesadores descritos con referencia a los métodos 300 y/o 400. En la etapa 505, el Procesador 1 puede ser ya consciente del tamaño de los datos a solicitar. El Procesador 1 puede ser consciente del tamaño de datos como resultado de otros protocolos, debido a un mensaje recibido previamente, porque un proceso relacionado que inicia la solicitud ha indicado el tamaño de datos, etc. El Procesador 1 puede asignar memoria asociada y/o fijar páginas de una manera similar a la etapa 303 en base al conocimiento previo del procesador del tamaño de los datos a solicitar. En la etapa 507, el Procesador 1 puede crear una lista de direcciones de destino para los datos de una manera similar a la etapa 305. En la etapa 509, el Procesador 1 puede transmitir un mensaje de escritura post al Procesador 2. El mensaje de escritura post de la etapa 509 puede comprender una solicitud para leer datos, una indicación de los datos a leer y la lista de direcciones de destino creada en la etapa 507. Como el mensaje de escritura post de la etapa 509 se puede referir a una transmisión desde el Procesador 2, el mensaje de escritura post de la etapa 509 se puede transmitir a la cola TX del Procesador 2. En la etapa 511, el Procesador 2 puede transmitir los datos solicitados a la(s) dirección(es) de destino en la dirección de destino por medio de DMA, mensajes “write post” adicionales, etc., de una manera similar a la etapa 309. En la etapa 513, el Procesador 2 puede transmitir un mensaje de escritura post que indica la finalización de la transferencia de una manera similar a la etapa 311. El mensaje de escritura post de la etapa 513 se puede transmitir a la cola de finalización del Procesador 1. Opcionalmente, el Procesador 1 puede transmitir un mensaje de escritura post de finalización a la cola de finalización del Procesador 2 en la etapa 515 de una manera similar a la etapa 313.

La FIG. 6 es un diagrama de protocolo de una forma de realización de un método de lectura de datos utilizando únicamente mensajes “write post” cuando se desconoce el tamaño de los datos. El método 600 se puede implementar con un procesador local (por ejemplo, un Procesador 1) que desea leer datos de un procesador remoto (por ejemplo, el Procesador 2), ambos de los cuales pueden ser, en el fondo, similares a los procesadores descritos con referencia a los métodos 300, 400 y/o 500. El método 600 puede ser, en el fondo, similar al método 500, pero se puede implementar cuando el Procesador 1 no es consciente del tamaño de los datos a solicitar. En la etapa 601, el

Procesador 1 puede transmitir un mensaje de escritura post que indique la intención de leer datos del Procesador 2 e identifique los datos a leer. Como el mensaje de escritura post de la etapa 601 se puede referir a una transmisión con el Procesador 2, el mensaje de escritura post de la etapa 601 se puede enrutar a la cola TX del Procesador 2. Una vez que el mensaje de la etapa 601 llega a la parte delantera de la cola TX, el Procesador 2 puede continuar a la etapa 603 y transmitir un mensaje de escritura post al Procesador 1 que indica el tamaño de los datos a leer. Como el mensaje de escritura post de la etapa 603 se puede referir a los datos a recibir por Procesador 1, el mensaje de la etapa 603 se puede enviar a la cola RX del Procesador 1. Una vez que el mensaje de la etapa 603 llega a la parte delantera de la cola RX, el Procesador 1 puede continuar con la etapa 605. Las etapas 605, 607, 609, 611, 613 y 615 pueden ser, en el fondo, similares a las etapas 505, 507, 509, 511, 513 y 515.

La FIG. 7 es un diagrama de flujo de otra forma de realización de un método 700 de lectura de datos utilizando únicamente mensajes “write post”. El método 700 se puede implementar con un procesador local (por ejemplo, un Procesador 1) que desee leer datos de un procesador remoto (por ejemplo, el Procesador 2), ambos de los cuales pueden ser, en el fondo, similares a los procesadores descritos con referencia a los métodos 300, 400, 500 y/o 600. En la etapa 701, el método 700 puede determinar si se conoce el tamaño de los datos a leer. El método 700 puede continuar a la etapa 707 si se conoce el tamaño de los datos y la etapa 703 si el tamaño de los datos se desconoce. En la etapa 703, se puede transmitir un mensaje de escritura post a una cola de transmisión en un procesador remoto. El mensaje de la etapa 703 puede indicar la intención de leer datos y solicitar información relacionada con un tamaño de los datos asociados. En la etapa 705, se puede recibir un mensaje de escritura post en una cola de recepción. El mensaje de la etapa 705 puede indicar la el tamaño de los datos solicitados. El método 700 puede a continuación continuar a la etapa 707. En la etapa 707, la memoria se puede asignar para recibir los datos en base al tamaño de los datos, se pueden fijar las páginas asociadas y se puede crear una SGL de direcciones de memoria asignadas. En la etapa 709, un mensaje de escritura post que comprende la SGL de direcciones de destino se puede transmitir a la cola de transmisión del procesador remoto. En la etapa 711, se pueden recibir el(los) mensaje(s) de escritura post y/o mensajes DMA que comprenden los datos solicitados en las direcciones de destino listadas en la SGL. En la etapa 713, se puede recibir un mensaje de escritura post en una cola de finalización y puede indicar que la transferencia de datos ha finalizado. Opcionalmente, en la etapa 715, se puede transmitir un mensaje de escritura post a una cola de finalización en el procesador remoto. El mensaje de escritura post de la etapa 715 puede indicar que los datos han sido recibidos completamente en las direcciones de destino.

REIVINDICACIONES

1. Un método de comunicación de datos a través de un puenteado no transparente (NTB) de una interconexión de componentes periféricos expreso (PCIe) (171) que comprende:

transmitir (301, 401) un primer mensaje “posted write” a un procesador remoto a través del NTB, en donde el primer mensaje “posted write” indica la intención de transferir los datos al procesador remoto; y

recibir (307, 403) un segundo mensaje “posted write” en respuesta al primer mensaje “posted write”, en donde el segundo mensaje “posted write” indica una lista de direcciones de destino para los datos.

2. El método de la reivindicación 1, que comprende además transmitir (309, 405) un tercer mensaje “posted write” a una dirección de destino obtenida de la lista de direcciones de destino del segundo mensaje “posted write”, en donde el tercer mensaje “posted write” comprende los datos.

3. El método de la reivindicación 1, que comprende además transmitir (309, 405) los datos a una dirección de destino obtenida de la lista de direcciones de destino del segundo mensaje “posted write” a través de acceso directo a memoria (DMA).

4. El método de la reivindicación 1, que comprende además transmitir (311, 407) un cuarto mensaje “posted write” al procesador remoto para indicar la finalización de una transferencia de los datos.

5. El método de la reivindicación 1, que comprende además recibir (313, 409) un quinto mensaje “posted write” del procesador remoto que indica la recepción completa de los datos.

6. Un método de comunicar datos a través de un puenteado no transparente (NTB) de una interconexión de componentes periféricos expreso (PCIe) que comprende:

transmitir (509, 609) un primer mensaje “posted write” a un procesador remoto a través del NTB, en donde el primer mensaje “posted write” comprende una solicitud de lectura de datos; y

recibir (511, 611) un mensaje de transferencia de datos que es otro mensaje “posted write” que comprende al menos algunos de los datos solicitados por el primer mensaje “posted write”.

7. El método de la reivindicación 6, en donde el primer mensaje “posted write” comprende una dirección de destino para los datos y en donde el mensaje de transferencia de datos se dirige a la dirección de destino.

8. El método de la reivindicación 7, que comprende además recibir (513, 613) un segundo mensaje “posted write” del procesador remoto que indica la finalización de una transferencia de datos asociada con la solicitud de lectura del primer mensaje “posted write”.

9. El método de la reivindicación 8, que comprende además transmitir (515, 615) un tercer mensaje “posted write” al procesador remoto que indica la completa recepción de los datos asociados con la solicitud de lectura del primer mensaje “posted write”.

10. Un procesador (115) que comprende:

un módulo de transferencia de datos (234) configurado para implementar una cola de recepción, una cola de transmisión y una cola de finalización;

en donde el procesador (115) se configura para:

acoplarse a un puenteado no transparente (NTB) de una interconexión de componentes periféricos expreso (PCIe) (171); y

leer datos de y escribir datos en, varios procesadores remotos a través de la cola de recepción, la cola de transmisión, la cola de finalización y el NTB PCIe (171) utilizando únicamente mensajes “posted write” y sin utilizar mensajes “non-posted”;

en donde el procesador se configura además para:

recibir mensajes “posted write” desde los varios procesadores remotos, indicando los mensajes “posted write” una lista de direcciones de destino para los datos;

almacenar los mensajes “posted write” en la cola de recepción, la cola de transmisión y la cola de finalización en base al contenido de los mensajes “posted write”.

11. El procesador de la reivindicación 10, en donde la cola de transmisión comprende una estructura de datos primero en entrar, primero en salir (FIFO), y en donde los mensajes “posted write” que indican solicitudes para transmitir datos se almacenan en la cola de transmisión.

12. El procesador de la reivindicación 10, en donde la cola de recepción comprende una estructura de datos primero en entrar, primero en salir (FIFO), y en donde los mensajes “posted write” que se dirigen el procesador para prepararse para recibir los datos se almacenan en la cola de recepción.

13. El procesador de la reivindicación 10, en donde la cola de finalización comprende una estructura de datos primero en entrar, primero en salir (FIFO), y en donde los mensajes “posted write” que indican que una transferencia de datos ha sido finalizada se almacenan en la cola de finalización.

14. El procesador de la reivindicación 10, en donde leer datos de y escribir datos en, los varios procesadores remotos a través de la cola de recepción, la cola de transmisión, la cola de finalización se realiza sin emplear un protocolo de acceso remoto directo a memoria (RDMA).

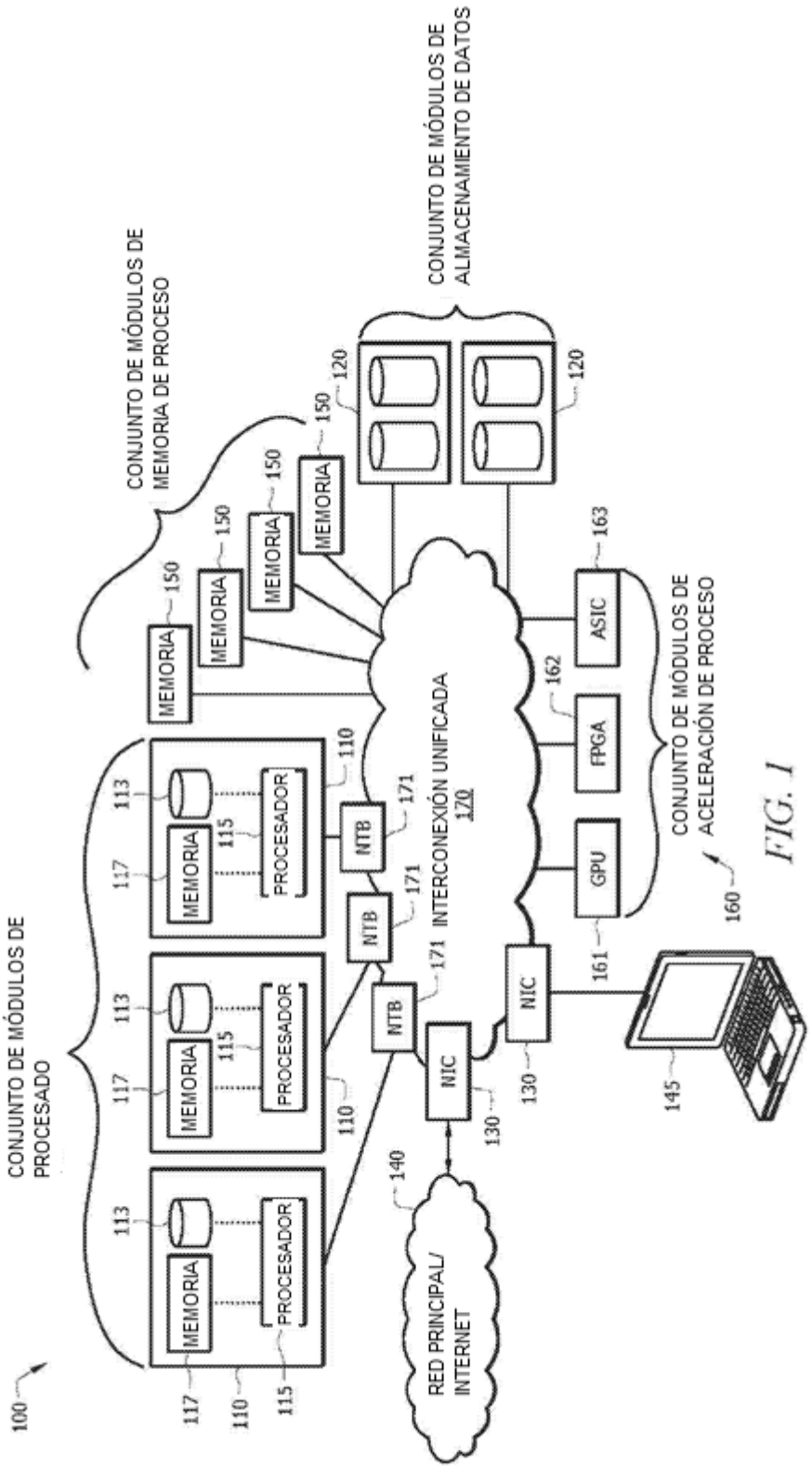
CONJUNTO DE MÓDULOS DE PROCESADO
CONJUNTO DE MÓDULOS DE MEMORIA DE PROCESO
CONJUNTO DE MÓDULOS DE ALMACENAMIENTO DE DATOS
CONJUNTO DE MÓDULOS DE ACELERACIÓN DE PROCESO
INTERCONEXIÓN UNIFICADA 170
RED PRINCIPAL/ INTERNET
MEMORIA
PROCESADOR
NTB
NIC
ASIC
FPGA
GPU
100
110
113
115
117
120
130
140
145
150
160
161
162
163
171

FIG. 1

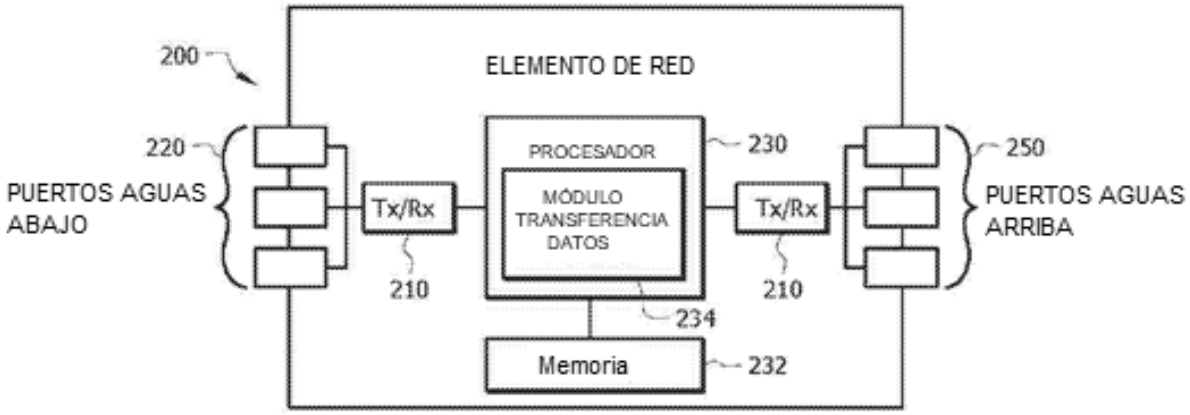
200
ELEMENTO DE RED
220
PUERTOS AGUAS ABAJO
Tx/Rx
210
PROCESADOR
230
MÓDULO TRANSFERENCIA DATOS
234
Memoria
232
Tx/Rx
210
250
PUERTOS AGUAS ARRIBA

FIG. 2

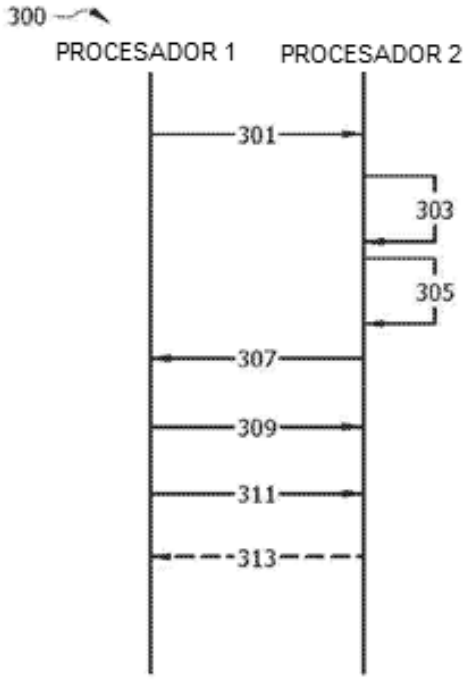
300
PROCESADOR 1
PROCESADOR 2
301
303
305
307
309
311
313

FIG. 3

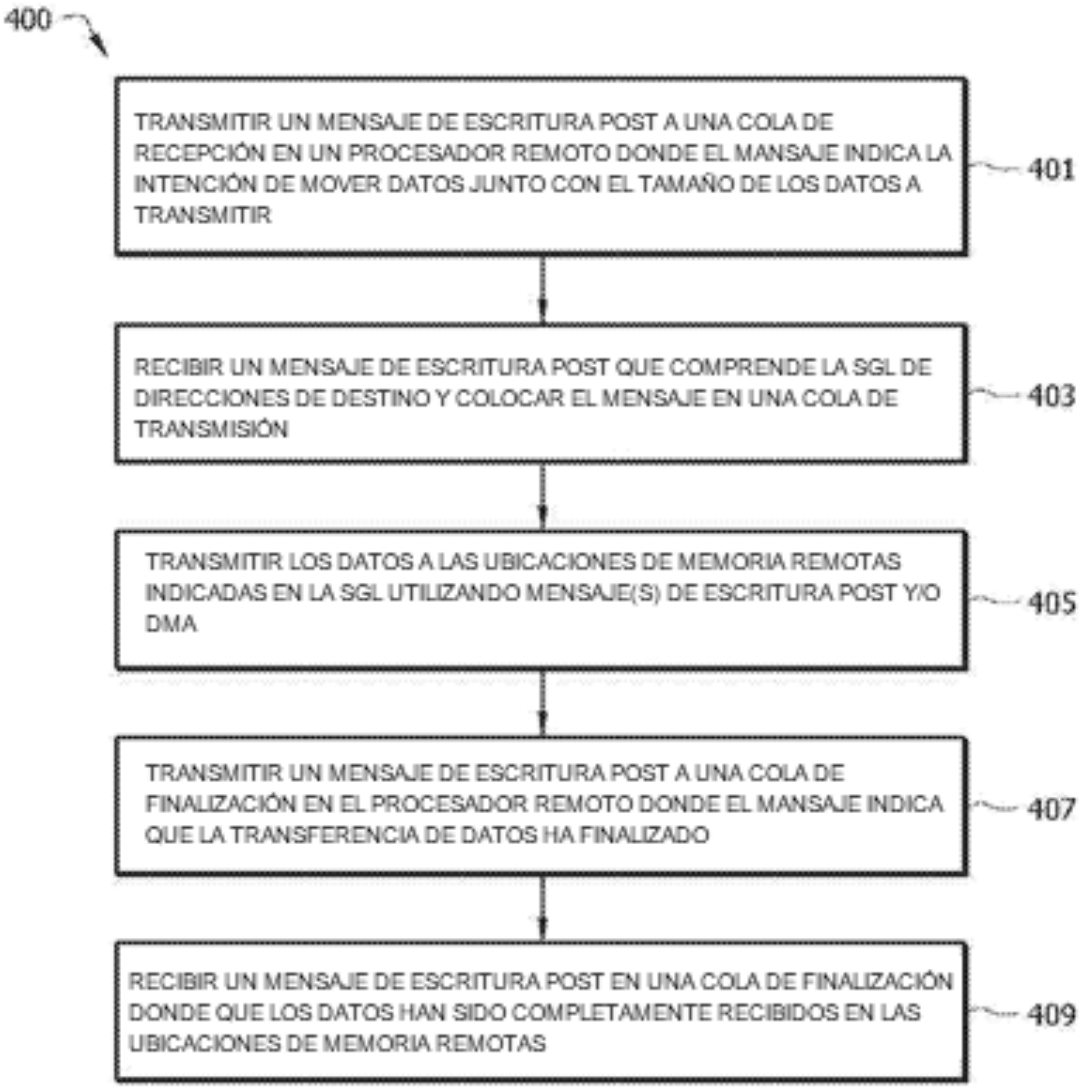
400
TRANSMITIR UN MENSAJE DE ESCRITURA POST A UNA COLA DE RECEPCIÓN EN UN PROCESADOR REMOTO DONDE EL MANSAJE INDICA LA INTENCIÓN DE MOVER DATOS JUNTO CON EL TAMAÑO DE LOS DATOS A TRANSMITIR
401
RECIBIR UN MENSAJE DE ESCRITURA POST QUE COMPRENDE LA SGL DE DIRECCIONES DE DESTINO Y COLOCAR EL MENSAJE EN UNA COLA DE TRANSMISIÓN
403
TRANSMITIR LOS DATOS A LAS UBICACIONES DE MEMORIA REMOTAS INDICADAS EN LA SGL UTILIZANDO MENSAJE(S) DE ESCRITURA POST Y/O DMA
405
TRANSMITIR UN MENSAJE DE ESCRITURA POST A UNA COLA DE FINALIZACIÓN EN EL PROCESADOR REMOTO DONDE EL MANSAJE INDICA QUE LA TRANSFERENCIA DE DATOS HA FINALIZADO
407
RECIBIR UN MENSAJE DE ESCRITURA POST EN UNA COLA DE FINALIZACIÓN DONDE QUE LOS DATOS HAN SIDO COMPLETAMENTE RECIBIDOS EN LAS UBICACIONES DE MEMORIA REMOTAS
409

FIG. 4

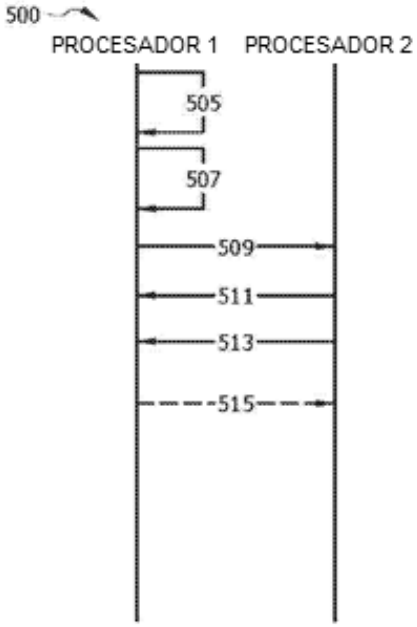
500
PROCESADOR 1
PROCESADOR 2
505
507
509
511
513
515

FIG. 5

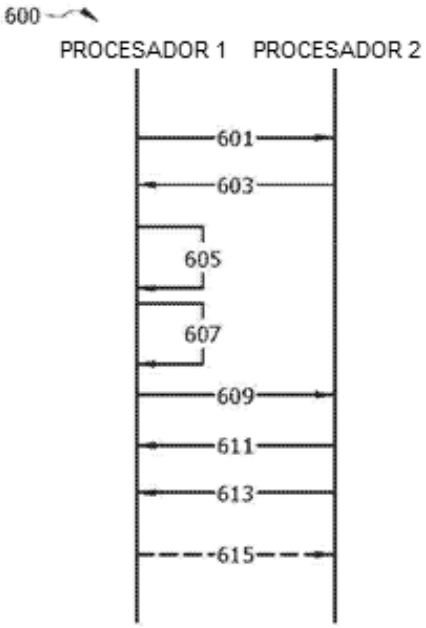
600
PROCESADOR 1
PROCESADOR 2
601
603
605
607
609
611
613
615

FIG. 6

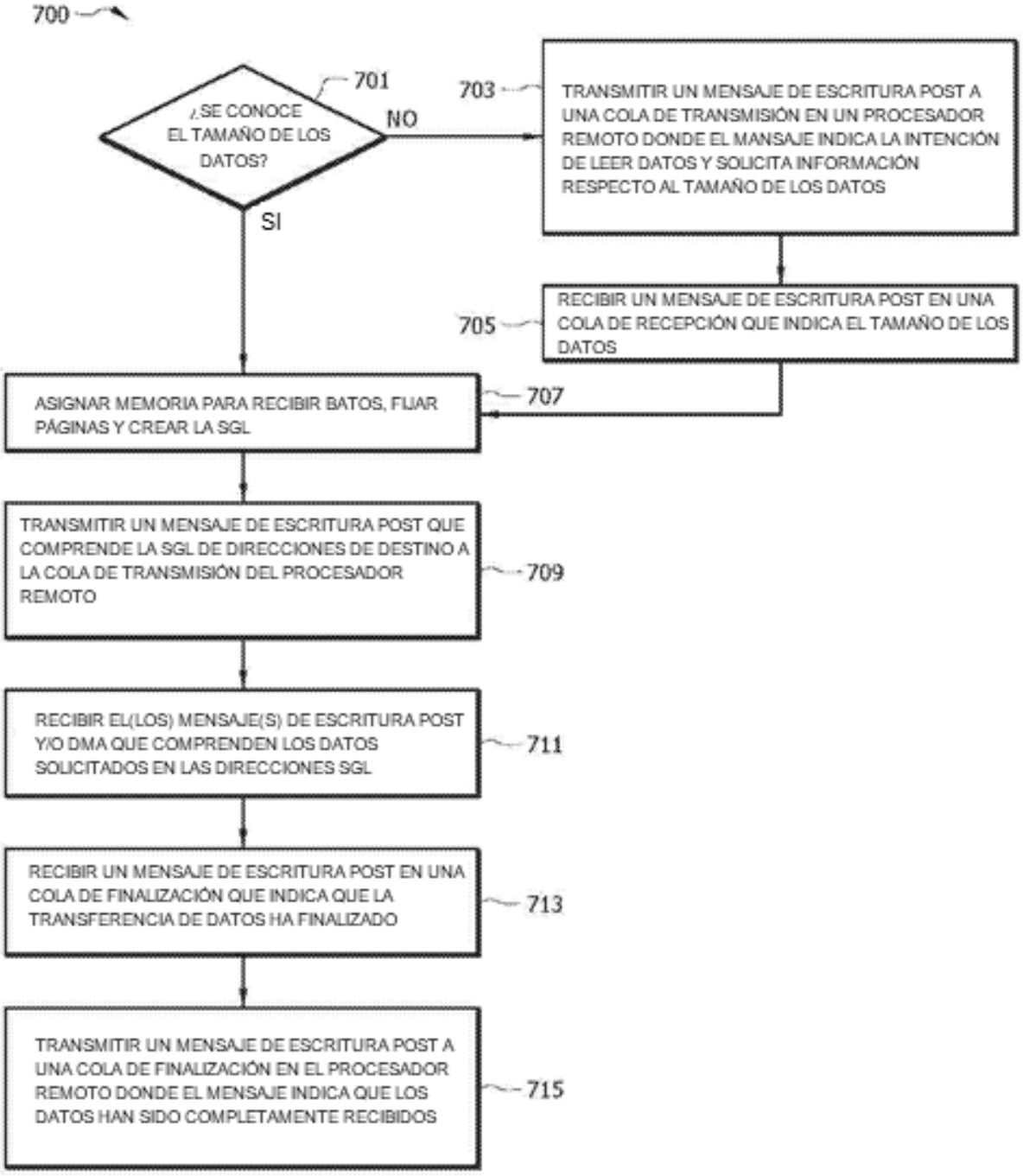
700
701
¿SE CONOCE EL TAMAÑO DE LOS DATOS?
NO
SI
703
TRANSMITIR UN MENSAJE DE ESCRITURA POST A UNA COLA DE TRANSMISIÓN EN UN PROCESADOR REMOTO DONDE EL MANSAJE INDICA LA INTENCIÓN DE LEER DATOS Y SOLICITA INFORMACIÓN RESPECTO AL TAMAÑO DE LOS DATOS
705
RECIBIR UN MENSAJE DE ESCRITURA POST EN UNA COLA DE RECEPCIÓN QUE INDICA EL TAMAÑO DE LOS DATOS
707
ASIGNAR MEMORIA PARA RECIBIR BATOS, FIJAR PÁGINAS Y CREAR LA SGL
709
TRANSMITIR UN MENSAJE DE ESCRITURA POST QUE COMPRENDE LA SGL DE DIRECCIONES DE DESTINO A LA COLA DE TRANSMISIÓN DEL PROCESADOR REMOTO
711
RECIBIR EL(LOS) MENSAJE(S) DE ESCRITURA POST Y/O DMA QUE COMPRENDEN LOS DATOS SOLICITADOS EN LAS DIRECCIONES SGL
713
RECIBIR UN MENSAJE DE ESCRITURA POST EN UNA COLA DE FINALIZACIÓN QUE INDICA QUE LA TRANSFERENCIA DE DATOS HA FINALIZADO
715
TRANSMITIR UN MENSAJE DE ESCRITURA POST A UNA COLA DE FINALIZACIÓN EN EL PROCESADOR REMOTO DONDE EL MENSAJE INDICA QUE LOS DATOS HAN SIDO COMPLETAMENTE RECIBIDOS

FIG. 7