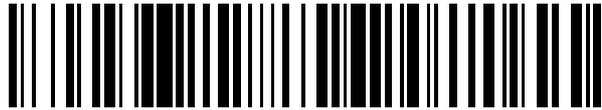


19



OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 703 473**

21 Número de solicitud: 201990003

51 Int. Cl.:

**G06F 15/16** (2006.01)  
**G06F 15/173** (2006.01)  
**G06F 9/46** (2006.01)  
**H04L 12/24** (2006.01)  
**H04L 29/08** (2006.01)

12

SOLICITUD DE PATENTE

A2

22 Fecha de presentación:  
**12.07.2017**

30 Prioridad:  
**12.07.2016 US 62/361,487**

43 Fecha de publicación de la solicitud:  
**08.03.2019**

71 Solicitantes:  
**PROXIMAL SYSTEMS CORPORATION (100.0%)**  
**2121 E. Kayland Way**  
**84117 Holladay UT Utah US**

72 Inventor/es:  
**INNES, Anthony y**  
**HILTON, Ronald**

74 Agente/Representante:  
**GARCÍA GONZÁLEZ, Sergio**

54 Título: **APARATO, SISTEMA Y PROCEDIMIENTO DE GESTIÓN DE ACOPLAMIENTO DE INTERMEDIARIO**

57 Resumen:

Aparato, sistema y procedimiento de gestión de acoplamiento de intermediario.

Se desvela un aparato, sistema y procedimiento para gestión de acoplamiento de intermediario. Un módulo 202 de plantilla de intermediario descarga 1014 de manera transparente una tarea 504 de un sistema 500-2 de procesamiento de datos nativo a una tarea 506 equivalente en un sistema 500-1 de procesamiento de datos remoto. Un módulo 204 de generación de intermediario rellena 810 la plantilla 202 de intermediario con información, tal como la especificación de una tarea 506 de descarga que es un equivalente remoto de la tarea 504 nativa, para generar un intermediario 508 de la tarea 504 nativa. Un módulo 206 de agente de descarga recibe una solicitud desde el intermediario 508 para realizar la tarea 506 de descarga en el sistema 500-1 de procesamiento de datos remoto.

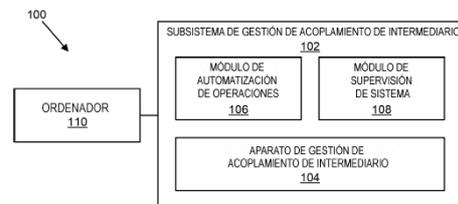


FIG. 1

## **DESCRIPCIÓN**

### **APARATO, SISTEMA Y PROCEDIMIENTO DE GESTIÓN DE ACOPLAMIENTO DE INTERMEDIARIO**

5

#### **REFERENCIA CRUZADA CON SOLICITUDES RELACIONADAS**

10 La presente solicitud reivindica el beneficio de la Solicitud de Patente Provisional de Estados Unidos N.º 62/361.487 titulada "APPARATUS, SYSTEM, AND METHOD FOR PROXY COUPLING MANAGEMENT" y presentada el 12 de julio de 2016 por Anthony D. Innes y Ronald N. Hilton, que se incorpora en el presente documento por referencia.

#### **ANTECEDENTES**

15

#### **CAMPO**

20 La materia objeto desvelada en el presente documento se refiere a acoplamiento de sistema cruzado y más particularmente se refiere a la descarga de tareas informáticas desde un procesador de datos o sistema de procesamiento a otro procesador de datos o sistema de procesamiento.

#### **DESCRIPCIÓN DE LA TÉCNICA RELACIONADA**

25 Como es conocido en la técnica, se emplean procesadores de descarga de fin especial para realizar ciertas tareas informáticas más eficazmente que un procesador de fin general o sistema de procesamiento. Tales procesadores se han implementado como un coprocesador fijado a un procesador de fin general que aumenta las capacidades de procesamiento del último para realizar operaciones  
30 realizadas tales como procesamiento de punto flotante, vectorial o criptográfico. Como alternativa, los procesadores de descarga pueden implementarse como pares de los procesadores de fin general en un sistema de múltiple procesamiento, con la capacidad para ejecutar tareas especializadas concurrentemente con otras tareas que se ejecutan en los procesadores de fin general. Un ejemplo de lo último serían  
35 los tipos de procesador zAAP y zIIP en un sistema de ordenador central z/arquitectura, que se ejecuta bajo el control del sistema operativo z/OS pero están confinados a ciertos tipos de tareas tales como miniaplicaciones de Java y consultas de base de datos.

En un diseño de procesador de descarga tradicional, los procesadores de fin general y de descarga todos se ejecutan dentro del mismo sistema de procesamiento de datos, como se define por la misma arquitectura global, y bajo el control de la misma gestión. Un diseño estrechamente acoplado tiende a minimizar la latencia de comunicación, pero también limita la flexibilidad y aumenta el coste fallando al aprovechamiento de la amplia diversidad de sistemas informáticos con diferentes arquitecturas y puntos de precio que están disponibles en el mercado hoy en día.

## 10 BREVE SUMARIO

Basándose en el análisis anterior, los inventores han reconocido una necesidad largamente detectada no satisfecha para un aparato, sistema, y procedimiento que descarguen sin interrupciones el procesamiento de tareas informáticas de un procesador de datos o sistema de procesamiento a otro procesador de datos o sistema de procesamiento de una arquitectura potencialmente diferente. De manera beneficiosa, un aparato, sistema, y procedimiento de este tipo mostrarían la flexibilidad y rentabilidad del acoplamiento de sistema cruzado mientras consiguen la transparencia y alta eficacia de procesadores de descarga estrechamente acoplados.

Las realizaciones de la presente invención se han desarrollado en respuesta al estado de la técnica actual, y en particular, en respuesta a los problemas y necesidades en la técnica que hasta ahora han probado ser intratables bajo procesadores de descarga actualmente disponibles. Por consiguiente, las realizaciones se han desarrollado para proporcionar un procedimiento, aparato, y sistema para procesamiento de descarga de un sistema de procesamiento de datos a otro sistema de procesamiento de datos de una arquitectura potencialmente diferente que supera muchas o todas las desventajas anteriormente analizadas en la técnica.

Cada sistema de procesamiento de datos puede incluir una pila de software que se ejecuta en uno o más procesadores, memoria, interfaces de dispositivo de E/S (Entrada/Salida), e interfaces de red, conectadas a través de un bus. Los sistemas de procesamiento de datos pueden consistir también en particiones lógicas o físicas separadas de un sistema de procesamiento de datos mayor, con algunos o todos los componentes de sistema anteriormente mencionados especializados a una partición específica o compartidos entre múltiples particiones de una manera virtualizada.

La tarea a descargarse se aloja en un primer sistema de procesamiento de

datos, y extiende sus servicios a un segundo sistema de procesamiento de datos mediante un intermediario que está alojado en el segundo sistema. La tarea y su intermediario cada uno se basan en su respectivo anfitrión local para servicios de sistema operativo básicos tales como despachar recursos de procesador, gestión de memoria, acceso de dispositivo de E/S e instalaciones para comunicar con el otro sistema.

En el nivel de aplicación, la tarea de descarga tiene la responsabilidad principal de las funciones internas de la aplicación, y la tarea de intermediario tiene la responsabilidad principal de comunicación externa con otras tareas relacionadas en el segundo sistema. La tarea de descarga y su intermediario comunican entre sí de una manera específica a las necesidades de la aplicación, operando de manera eficaz como rutinas conjuntas que comprenden una única tarea lógica.

Cuanto más eficaces sean las instalaciones subyacentes para comunicación inter-sistema mayor será la eficacia del mecanismo global que puede acercarse a la de los mecanismos de descarga estrechamente acoplados en la técnica anterior, pero sin el coste añadido y la inflexibilidad asociada con tales mecanismos de fin especial.

Obsérvese que el papel del primer y segundo sistema de procesamiento de datos es dependiente de la tarea. Un sistema puede servir como el sistema de descarga para una tarea mientras que sirve simultáneamente como el sistema de intermediario para otra tarea. Este enfoque para procesamiento de datos de descarga entre sistemas acoplados cruzados se denomina acoplamiento de intermediario, que incluye un aparato, sistema y procedimiento para gestionarlo.

El aparato para gestionar acoplamiento de intermediario se proporciona con una pluralidad de módulos configurados para ejecutar las etapas necesarias para proporcionar una plantilla de intermediario para descargar de manera transparente una tarea de un sistema de procesamiento de datos nativo a una tarea equivalente en un sistema de procesamiento de datos remoto, rellenar la plantilla de intermediario con información, tal como la especificación de una tarea de descarga que es un equivalente remoto de la tarea nativa, para generar un intermediario de la tarea nativa, y recibir una solicitud del intermediario para realizar la tarea de descarga en el sistema de procesamiento de datos remoto. Estos módulos en las realizaciones descritas incluyen un módulo de plantilla de intermediario, un módulo de generación de intermediario, y un módulo de agente de descarga.

El módulo de plantilla de intermediario puede supervisar también el estado del agente de descarga, cancelar la solicitud al agente de descarga, e iniciar una conmutación por recuperación para revertir a la tarea nativa. El módulo de generación de intermediario puede también rellenar información adicional, tal como

la especificación de datos de entrada y salida que necesita la tarea nativa, o un alias de la tarea nativa a invocarse para los fines de conmutación por recuperación automática en el caso de un error. El módulo de agente de descarga puede también poner en cola una solicitud de descarga desde el intermediario a una cola pendiente, mover la solicitud desde la cola pendiente a una cola activa y activar la correspondiente tarea de descarga, y supervisar la tarea de descarga en la cola activa y quitarla de la cola tras la finalización.

En una realización, el aparato incluye un módulo de configuración de intermediario que incorpora el intermediario y su correspondiente tarea de descarga en una base de datos de gestión de configuración (CMDB), mejora seguridad y control de acceso para código y datos de intermediario y ajusta parámetros de control del intermediario y activa inicialmente el intermediario. El aparato puede incluir también un módulo de instalación de intermediario que crea parámetros de control para el intermediario y asocia un alias de tarea nativa con el intermediario.

En una realización adicional, el aparato incluye un módulo de conmutación por recuperación que revierte de vuelta a la tarea nativa si el intermediario, el módulo de agente de descarga, o el sistema de procesamiento de datos remoto experimentan una interrupción, ajusta la capacidad del sistema de procesamiento de datos nativo, desactiva cualquier intermediario afectado, y restaura la operación normal cuando se ha resuelto la interrupción.

También se presenta un sistema para gestionar acoplamiento de intermediario. El sistema puede realizarse para incluir un primer/remoto y un segundo/nativo sistema de procesamiento de datos, un mecanismo de comunicación, el anterior aparato de gestión de acoplamiento de intermediario, y un sistema de almacenamiento para compartir datos de entrada y salida. En particular, el sistema, en una realización, incluye un módulo de automatización de operaciones y un módulo de supervisión de sistema.

En una realización, el módulo de conmutación por recuperación puede informar al módulo de automatización de operaciones cuando ha tenido lugar una conmutación por recuperación. En otra realización, el módulo de agente de descarga puede informar al módulo de supervisión de sistema de datos de medición solicitados.

También se presenta un procedimiento para gestión de acoplamiento de intermediario. El procedimiento en las realizaciones desveladas sustancialmente incluye las etapas necesarias para llevar a cabo las funciones anteriormente presentadas con respecto a la operación del aparato y sistema descritos.

Las referencias a través de toda esta memoria descriptiva a características, ventajas o lenguaje similar no implican que todas las características y ventajas

puedan realizarse en alguna única realización. En su lugar, el lenguaje que hace referencia a las características y ventajas se entiende para significar que un rasgo, ventaja o característica específica se incluye en al menos una realización. Por lo tanto, el análisis de las características y ventajas, y lenguaje similar, a través de toda esta memoria descriptiva puede hacer referencia, pero no necesariamente, a la misma realización.

Adicionalmente, los rasgos, ventajas y características descritas de las realizaciones pueden combinarse de cualquier manera adecuada. Un experto en la materia reconocerá que las realizaciones pueden ponerse en práctica sin una o más de las características específicas o ventajas de una realización particular. En otras instancias, pueden reconocerse características y ventajas adicionales en ciertas realizaciones que pueden no estar presentes en todas las realizaciones.

Estas características y ventajas de las realizaciones se harán más completamente evidentes a partir de la siguiente descripción y reivindicaciones adjuntas, o pueden aprenderse por la puesta en práctica de las realizaciones como se expone en lo sucesivo.

## **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

Se presentará una descripción más particular de las realizaciones brevemente descritas anteriormente por referencia a realizaciones específicas que se ilustran en los dibujos adjuntos. Entendiendo que estos dibujos representan únicamente algunas realizaciones y no se han de considerar por lo tanto que son limitantes del alcance, las realizaciones se describirán y explicarán con especificidad y detalle adicional a través del uso de los dibujos adjuntos, en los que:

La Figura 1 es un diagrama de bloques esquemático que ilustra un sistema de la presente invención;

La Figura 2 es un diagrama de bloques esquemático que ilustra un aparato de gestión de acoplamiento de intermediario;

La Figura 3 es un diagrama de bloques esquemático que ilustra una posible plataforma de hardware informático en la que puede desplegarse la presente invención al menos en parte;

La Figura 4 es un diagrama de bloques esquemático de un posible ordenador que incluye una pila de software en la que puede residir la presente invención al menos en parte;

La Figura 5 es un diagrama de bloques esquemático de dos ordenadores que operan de acuerdo con la presente invención;

La Figura 6 es un diagrama de bloques esquemático de una posible

configuración de hardware de sistemas de múltiples procesamientos de datos en los que puede residir al menos parcialmente la presente invención;

La Figura 7 es un diagrama de flujo que ilustra una realización de configuración de intermediario;

5 La Figura 8 es un diagrama de flujo que ilustra una realización de generación de intermediario;

La Figura 9 es un diagrama de flujo que ilustra una realización de instalación de intermediario, después de que se haya generado o recuperado un intermediario de la biblioteca de intermediarios;

10 La Figura 10 es un diagrama de flujo que ilustra una realización de una plantilla de intermediario;

Las Figuras 11A-11D son diagramas de flujo que ilustran una realización de un agente de descarga; y

15 La Figura 12 es un diagrama de flujo que ilustra una realización de conmutación por recuperación .

## **DESCRIPCIÓN DETALLADA**

20 Como se apreciará por un experto en la materia, los aspectos de las realizaciones pueden realizarse como un sistema, procedimiento o producto de programa informático. Por consiguiente, las realizaciones pueden tomar la forma de una realización completamente de hardware, una realización completamente de software (que incluye soporte lógico inalterable, software residente, micro-código, etc.) o una realización que combina aspectos de software y hardware que pueden  
25 todos denominarse en general en el presente documento como un “circuito”, “módulo” o “sistema”. Adicionalmente, las realizaciones pueden tomar la forma de un producto de programa informático realizado en uno o más medio o medios legibles por ordenador que tienen código de programa legible por ordenador incorporado en los mismos.

30 Muchas de las unidades funcionales descritas en esta memoria descriptiva se han etiquetado como módulos, para destacar más particularmente su independencia de implementación. Por ejemplo, un módulo puede implementarse como un circuito de hardware que comprende circuitos de VLSI personalizados o matrices de puertas, semiconductores existentes tales como chips lógicos, transistores u otros  
35 componentes discretos. Un módulo puede implementarse también en dispositivos de hardware programable tales como campos de matrices de puertas programables, lógica de matrices programable, dispositivos lógicos programables o similares.

Los módulos pueden implementarse también en software para ejecución por

diversos tipos de procesadores. Un módulo identificado de código de programa legible por ordenador puede comprender, por ejemplo, uno o más bloques físicos o lógicos de instrucciones informáticas que pueden organizarse, por ejemplo, como un objeto, procedimiento o función. Sin embargo, los ejecutables de un módulo  
5 identificado no necesitan estar físicamente localizados juntos, sino que pueden comprender instrucciones separadas almacenadas en diferentes localizaciones que, cuando se unen lógicamente juntas, comprenden el módulo y consiguen el fin establecido por el módulo.

De hecho, un módulo de código de programa legible por ordenador puede ser  
10 una única instrucción, o muchas instrucciones, y puede incluso distribuirse a través de varios diferentes segmentos de código, entre diferentes programas, y a través de varios dispositivos de memoria. De manera similar, pueden identificarse e ilustrarse datos operacionales en el presente documento dentro de los módulos, y pueden realizarse en cualquier forma adecuada y organizarse dentro de cualquier tipo  
15 adecuado de estructura de datos. Los datos operacionales pueden recopilarse como un único conjunto de datos, o pueden distribuirse a través de diferentes localizaciones incluyendo a través de diferentes dispositivos de almacenamiento, y pueden existir, al menos parcialmente, simplemente como señales electrónicas en un sistema o red. Donde un módulo o porciones de un módulo se implementan en  
20 software, las porciones de software se almacenan en uno o más medio o medios legibles por ordenador.

Puede utilizarse cualquier combinación de uno o más medio o medios legibles por ordenador. El medio legible por ordenador puede ser un medio de señal legible por ordenador o un medio de almacenamiento legible por ordenador. El medio  
25 legible por ordenador puede ser un medio de almacenamiento legible por ordenador tangible que almacena el código legible por ordenador. El medio de almacenamiento legible por ordenador puede ser, por ejemplo, pero sin limitación, un sistema, aparato o dispositivo electrónico, magnético, óptico, electromagnético, de infrarrojos, holográfico, micromecánico o de semiconductores o cualquier combinación  
30 adecuada de lo anterior.

Ejemplos más específicos (una lista no exhaustiva) del medio legible por ordenador incluirían los siguientes: una conexión eléctrica que tiene uno o más cables, un disquete de ordenador portátil, un disco duro, una memoria de acceso aleatorio (RAM), una memoria de solo lectura (ROM), una memoria de solo lectura  
35 borrrable programable (EPROM o memoria Flash), una memoria de solo lectura de disco compacto portátil (CD-ROM), un dispositivo de almacenamiento óptico, un dispositivo de almacenamiento magnético, o cualquier combinación adecuada de lo anterior. En el contexto de este documento, un medio de almacenamiento legible por

ordenador puede ser cualquier medio tangible que pueda contener, o almacenar un programa para su uso por o en relación con un sistema, aparato o dispositivo de ejecución de instrucciones.

5 Un medio de señal legible por ordenador puede incluir una señal de datos propagada con código de programa legible por ordenador incorporada en el mismo, por ejemplo, en banda base o como parte de una onda portadora. Una señal propagada de este tipo puede tomar cualquiera de una diversidad de formas, incluyendo, pero sin limitación, electromagnética, óptica o cualquier combinación adecuada de las mismas. Un medio de señal legible por ordenador puede ser  
10 cualquier medio legible por ordenador que no sea un medio de almacenamiento legible por ordenador y que pueda comunicar, propagar o transportar un programa para su uso por o en relación con un sistema, aparato o dispositivo de ejecución de instrucciones. Código de programa legible por ordenador realizado en un medio legible por ordenador puede transmitirse usando cualquier medio apropiado,  
15 incluyendo, pero sin limitación, inalámbrico, alámbrico, cable de fibra óptica, RF, etc., o cualquier combinación adecuada de lo anterior.

Código de programa legible por ordenador para llevar a cabo operaciones para las realizaciones puede escribirse en cualquier combinación de uno o más lenguajes de programación, incluyendo un lenguaje de programación orientado a  
20 objetos tal como Java, Smalltalk, C++ o similares y lenguajes de programación procedural convencionales, tal como el lenguaje de programación "C" o lenguajes de programación similares. El código de programa legible por ordenador puede ejecutarse completamente en el ordenador del usuario, parcialmente en el ordenador del usuario, como un paquete de software independiente, parcialmente en el  
25 ordenador del usuario y parcialmente en un ordenador remoto o completamente en el ordenador remoto o servidor. En el último escenario, el ordenador remoto puede estar conectado al ordenador del usuario a través de un tipo de red, incluyendo una red de área local (LAN) o una red de área extensa (WAN), o la conexión puede realizarse a un ordenador externo (por ejemplo, a través de la Internet usando un  
30 Proveedor de Servicio de Internet).

La referencia a través de toda esta memoria descriptiva a "una realización" o lenguaje similar significa que un rasgo, estructura o característica particular descrita en relación con la realización está incluida en al menos una realización. Por lo tanto, las apariciones de las frases "en una realización" y lenguaje similar a través de toda  
35 esta memoria descriptiva pueden hacer todas referencia, pero no necesariamente, a la misma realización, pero significan "una o más pero no todas las realizaciones" a menos que se especifique expresamente de lo contrario. Las expresiones "que incluye", "que comprende", "que tiene" y variaciones de las mismas significan "que

incluye pero sin limitación,” a menos que se especifique expresamente de lo contrario. Una lista enumerada de elementos no implica que cualquiera o todos los elementos sean mutuamente exclusivos, a menos que se especifique expresamente de lo contrario. Los términos “un”, “una”, “el” y “la” también hacen referencia a “uno o  
5 más” a menos que se especifique expresamente de lo contrario.

Adicionalmente, los rasgos descritos, estructuras, o características de las realizaciones pueden combinarse de cualquier manera adecuada. En la siguiente descripción, se proporcionan numerosos detalles específicos, tales como ejemplos de programación, módulos de software, selecciones de usuario, transacciones de red, consultas de base de datos, estructuras de base de datos, módulos de  
10 hardware, circuitos de hardware, chips de hardware, etc., para proporcionar un entendimiento minucioso de las realizaciones. Un experto en la materia reconocerá, sin embargo, que las realizaciones pueden ponerse en práctica sin uno o más de los detalles específicos o con otros procedimientos, componentes, materiales y así sucesivamente. En otras instancias, estructuras bien conocidas, materiales u  
15 operaciones no se muestran o se describen en detalle para evitar oscurecer aspectos de una realización.

Los aspectos de las realizaciones se describen a continuación con referencia a diagramas de flujo esquemáticos y/o diagramas de bloques esquemáticos de los procedimientos, aparatos, sistemas, y productos de programa informático de acuerdo con las realizaciones. Se entenderá que cada bloque de los diagramas de flujo esquemáticos y/o diagramas de bloques esquemáticos, y combinaciones de bloques en los diagramas de flujo esquemáticos y/o diagramas de bloques esquemáticos, pueden implementarse por código de programa legible por  
20 ordenador. Este código de programa legible por ordenador puede proporcionarse a un procesador de un ordenador de fin general, ordenador de fin especial, u otro aparato de procesamiento de datos programable para producir una máquina, de manera que las instrucciones, que se ejecutan mediante el procesador del ordenador u otro aparato de procesamiento de datos programable, crean medios  
25 para implementar las funciones/actos especificados en los diagramas de flujo esquemáticos y/o bloque o bloques de diagramas de bloques esquemáticos.

El código de programa legible por ordenador puede también almacenarse en un medio legible por ordenador que puede dirigir un ordenador, otro aparato de procesamiento de datos programable, u otros dispositivos para funcionar de una  
35 manera particular, de manera que las instrucciones almacenadas en el medio legible por ordenador producen un artículo de fabricación que incluye instrucciones que implementan la función/acto especificado en los diagramas de flujo esquemáticos y/o bloque o bloques de diagramas de bloques esquemáticos.

El código de programa legible por ordenador puede cargarse también en un ordenador, otro aparato de procesamiento de datos programable, u otros dispositivos para provocar que se realicen una serie de etapas opcionales en el ordenador, otro aparato programable u otros dispositivos para producir un proceso  
 5 implementado por ordenador de manera que el código de programa que se ejecuta en el ordenador u otro aparato programable proporciona procesos para implementar las funciones/actos especificados en el diagrama de flujo y/o bloque o bloques de diagramas de bloques esquemáticos.

Los diagramas de flujo esquemáticos y/o diagramas de bloques esquemáticos  
 10 en las figuras ilustran la arquitectura, funcionalidad, y operación de posibles implementaciones de aparatos, sistemas, procedimientos y productos de programa informático de acuerdo con diversas realizaciones. En este sentido, cada bloque en los diagramas de flujo esquemáticos y/o diagramas de bloques esquemáticos pueden representar un módulo, segmento, o porción de código, que comprende una  
 15 o más instrucciones ejecutables del código de programa para implementar la función o funciones lógicas especificadas.

Debería observarse también que, en algunas implementaciones alternativas, las funciones indicadas en el bloque pueden tener lugar fuera del orden indicado en las figuras. Por ejemplo, dos bloques mostrados en sucesión pueden ejecutarse, de  
 20 hecho, sustancialmente de manera concurrente, o los bloques pueden ejecutarse en ocasiones en el orden inverso, dependiendo de la funcionalidad implicada. Pueden concebirse otras etapas y procedimientos que son equivalentes en función, lógica o efecto a uno o más bloques, o porciones de los mismos de las figuras ilustradas.

Aunque pueden emplearse diversos tipos de flechas y líneas en el diagrama  
 25 de flujo y/o diagramas de bloques, se entiende que no limitan el alcance de las correspondientes realizaciones. De hecho, algunas flechas u otros conectores pueden usarse para indicar únicamente el flujo lógico de la realización representada. Por ejemplo, una flecha puede indicar un periodo de espera o supervisión de duración no especificada entre etapas enumeradas de la realización representada.  
 30 Se observará también que cada bloque de los diagramas de bloques y/o diagramas de flujo, y combinaciones de bloques en los diagramas de bloques y/o diagramas de flujo, puede implementarse por sistemas basados en hardware de fin especial que realizan las funciones o actos especificados, o combinaciones de hardware de fin especial y código de programa legible por ordenador.

35 La Figura 1 es un diagrama de bloques esquemático que ilustra un sistema 100 de una realización actual, que incluye un ordenador o sistema 110 de manejo de información y un subsistema 102 de gestión de acoplamiento de intermediario. El subsistema 102 incluye adicionalmente un aparato 104 de gestión de acoplamiento

de intermediario, un módulo 106 de automatización de operaciones, y un módulo 108 de supervisión de sistema. En una realización, los componentes anteriores del subsistema 102 pueden implementarse completa o parcialmente dentro de una plataforma de hardware y/o una pila de software del ordenador 110.

5 El aparato 104 puede proporcionar una plantilla de intermediario, rellenar la plantilla de intermediario con información para generar un intermediario de la tarea nativa, y recibir una solicitud desde el intermediario para realizar la tarea de descarga.

10 El módulo 106 de automatización de operaciones puede estar informado cuando tiene lugar una conmutación por recuperación y ayudar con la recuperación. El módulo 108 de supervisión de sistema puede estar informado de datos de medición solicitados. En una realización, puede incluirse una indicación de tiempo.

15 La Figura 2 es un diagrama de bloques esquemático que ilustra un aparato 104 de gestión de acoplamiento de intermediario, que incluye un módulo 202 de plantilla de intermediario, un módulo 204 de generación de intermediario, un módulo 206 de agente de descarga, un módulo 208 de configuración de intermediario, un módulo 210 de instalación de intermediario, y un módulo 212 de conmutación por recuperación . El módulo 202 de plantilla de intermediario puede descargar de manera transparente una tarea de un sistema 500-2 de procesamiento de datos nativo a una tarea equivalente en un sistema 500-1 de procesamiento de datos remoto. El módulo 204 de generación de intermediario puede rellenar la plantilla de intermediario con información, tal como la especificación de una tarea de descarga que es un equivalente remoto de la tarea nativa, para generar un intermediario de la tarea nativa. El módulo 206 de agente de descarga puede recibir una solicitud desde el intermediario para realizar la tarea de descarga en el sistema 500-1 de procesamiento de datos remoto. El módulo 208 de configuración de intermediario puede incorporar el intermediario y su correspondiente tarea de descarga en una base de datos de gestión de configuración (CMDB). El módulo 210 de instalación de intermediario puede crear parámetros de control para el intermediario. El módulo 212 de conmutación por recuperación puede revertir de vuelta a la tarea nativa si el intermediario, el módulo de agente de descarga, o el sistema 500-1 de procesamiento de datos remoto experimentan una interrupción.

25 El módulo 202 de plantilla de intermediario puede supervisar adicionalmente el estado del agente de descarga. En diversas realizaciones, el módulo 202 de plantilla de intermediario puede cancelar también la solicitud al agente de descarga. En una realización adicional, el módulo 202 de plantilla de intermediario puede iniciar una conmutación por recuperación para revertir a la tarea nativa.

El módulo 204 de generación de intermediario puede rellenar adicionalmente

información adicional, tal como la especificación de datos de entrada y salida que necesita la tarea nativa, o un alias de la tarea nativa a invocarse para los fines de conmutación por recuperación automática en el caso de un error. El módulo 206 de agente de descarga puede poner en cola adicionalmente una solicitud de descarga desde el intermediario a una cola pendiente, mover la solicitud desde la cola pendiente a una cola activa y activar la correspondiente tarea de descarga, y supervisar la tarea de descarga en la cola activa y quitarla de la cola tras la finalización. El módulo 208 de configuración de intermediario puede mejorar adicionalmente la seguridad y control de acceso para código y datos de intermediario y ajustar parámetros de control del intermediario e iniciar la activación del intermediario. El módulo 210 de instalación de intermediario puede asociar adicionalmente un alias de tarea nativa con el intermediario. El módulo 212 de conmutación por recuperación puede ajustar adicionalmente la capacidad del sistema 500-2 de procesamiento de datos nativo, desactivar cualquier intermediario afectado y restaurar la operación de acoplamiento de intermediario normal cuando se ha resuelto la interrupción.

La Figura 3 ilustra una posible plataforma 300 de hardware informático en la que puede desplegarse la presente invención al menos en parte. La plataforma 300 de hardware puede incluir el procesador o procesadores 302, memoria 304, una interfaz 306 de red, y una interfaz 308 de dispositivo de E/S (Entrada/Salida), conectada a través de un bus 310.

La plataforma 300 de hardware puede ser de cualquier factor de forma o tipo, incluyendo un sistema embebido, uno portable, un portátil, un ordenador personal, un miniordenador, un servidor, un ordenador central, un superordenador, y similares.

El procesador o procesadores 302 pueden estar presentes en cualquier cantidad, incluyendo un único procesador, y puede tener cualquier arquitectura de conjunto de instrucciones. En una realización, el procesador o procesadores 302 pueden tener uno o más niveles de cachés especializadas o compartidas. Las posibles implementaciones físicas pueden incluir múltiples chips, chip único, múltiples núcleos, procesadores con tecnología Hyper Threading y similares.

La memoria 304 puede ser de cualquier tamaño u organización y puede incluir tanto secciones de lectura/escritura como de solo lectura. Puede incluir también tanto secciones globales como locales, y puede soportar tanto acceso uniforme como no uniforme. Puede incorporar E/S de memoria mapeada y acceso a memoria directo. Puede soportar coherencia de caché, incluyendo protocolos basados en directorio y de tipo espía.

La interfaz 306 de red puede soportar cualquier protocolo o de red o arquitectura. Puede soportar tanto conexiones de red inalámbricas como de

cableado permanente. Puede comprender Ethernet, Anillo con Paso de Testigo, Arquitectura de Red de Sistema ("SNA"), y similares. En una realización, puede estar integrado con la interfaz 308 de dispositivo de E/S.

5 La interfaz 308 de dispositivo de E/S puede controlarse principalmente por el procesador o procesadores 302 o puede incorporar un subsistema de procesador de E/S independiente. Puede comprender Interconexión de Componentes Periféricos ("PCI"), Interfaz de Sistema Informático Pequeña ("SCSI"), Canal de Fibra ("FC"), Conexión de Sistema Empresarial ("ESCON"), ESCON sobre Canal de Fibra ("FICON"), y similares. En una realización, puede incluir dispositivos de E/S locales  
10 especializados.

El bus 310 puede comprender uno o más de una diversidad de topologías físicas y lógicas. Puede ser paralelo o serie. Puede ser unidireccional o bidireccional. Puede ser plano o jerárquico. Puede comprender una barra transversal completa o parcial. Puede comprender múltiples buses en puente. En una realización, el bus  
15 310 puede comprender una red interna de alta velocidad.

La Figura 4 es un diagrama de un posible ordenador 110 que incluye una pila de software en la que puede residir la presente invención al menos en parte. La pila de software puede incluir la tarea o tareas 402, alojadas en un sistema operativo 404, posibilitado por soporte lógico inalterable 406, que se ejecuta en una plataforma  
20 300 de hardware de la cual la configuración de Figure 3 es representativa.

La tarea o tareas 402 pueden incluir tanto tareas de nivel de usuario como de sistema. Pueden ser interactivas o en lotes. Pueden ejecutarse en primer plano o en segundo plano. La tarea o tareas 402 de nivel de usuario pueden incluir aplicaciones, programas, trabajos, soporte intermedio, y similares. La tarea o tareas  
25 402 de nivel de sistema pueden incluir servicios, controladores, demonios, utilidades, y similares.

El sistema operativo 404 puede ser de cualquier tipo y versión y en cualquier estado. Los tipos pueden incluir Unix, Linux, Windows, Mac, MVS, VMS, y similares. Las versiones pueden incluir Windows XP, Windows Vista, y similares. Los estados  
30 pueden incluir un grado de personalización, un modo de operación, una preparación de sistema para configuración y similares. El sistema operativo 404 puede ser de único usuario o de múltiples usuarios. Puede ser mono-tarea o multi-tarea. En una realización, el sistema operativo 404 puede ser en tiempo real. En otra realización, el sistema operativo 404 puede estar embebido.

35 El soporte lógico inalterable 406 puede comprender microcódigo, que puede residir en un microalmacén del procesador o procesadores 302. En una realización, el soporte lógico inalterable 406 puede comprender software de bajo nivel, que puede residir en memoria 304. En una realización, el soporte lógico inalterable 406

puede comprender un sistema operativo 404 rudimentario. En una realización adicional, el soporte lógico inalterable 406 puede soportar virtualización para permitir la operación concurrente de múltiples sistemas operativos 404 en una plataforma 300 de hardware.

5           La Figura 5 es un diagrama de bloques esquemático de dos ordenadores 500 que incluyen sus respectivas pilas de software que operan de acuerdo con la presente invención. La primera y segunda pilas de software pueden incluir respectivamente la primera tarea o tareas 502 y la segunda tarea o tareas 504 que pueden o pueden no diferir en cuanto a número y función, alojadas respectivamente  
10 en un primer sistema operativo 510 y en un segundo sistema operativo 512 potencialmente diferente, activadas respectivamente por el primer soporte lógico inalterable 514 y por un segundo soporte lógico inalterable 516 potencialmente diferente, y que se ejecutan respectivamente en una primera plataforma 518 de hardware y en una segunda plataforma 520 de hardware potencialmente diferente.  
15 Dichas plataformas de hardware pueden ser también particiones lógicas o físicas de una o más plataformas de hardware más grandes.

De particular relevancia para la presente invención son la tarea 506 de descarga y la tarea 508 de intermediario. Estas tareas están alojadas respectivamente en la primera y segunda pilas de software que se ejecutan  
20 respectivamente en la primera y segunda plataformas 518 y 520 de hardware, en lo sucesivo denominadas respectivamente como un primer sistema 500-1 de procesamiento de datos y un segundo sistema 500-2 de procesamiento de datos. La tarea 506 de descarga y su correspondiente tarea 508 de intermediario cada una están basadas en su respectivo anfitrión local, el primer sistema 500-1 de  
25 procesamiento de datos y el segundo sistema 500-2 de procesamiento de datos, para servicios de sistema operativo básicos tales como despachar recursos de procesador, gestión de memoria, acceso de dispositivo de E/S, y facilita la comunicación con el otro sistema.

La tarea 506 de descarga en el primer sistema 500-1 de procesamiento de  
30 datos extiende sus servicios al segundo sistema 500-2 de procesamiento de datos mediante la tarea 508 de intermediario. En el nivel de aplicación (o soporte intermedio o controlador), la tarea 506 de descarga tiene la responsabilidad principal de las funciones internas de la aplicación, y la tarea 508 de intermediario tiene la responsabilidad principal de comunicación externa con otras segundas tareas  
35 relacionadas 504 en el segundo sistema 500-2 de procesamiento de datos. La tarea 506 de descarga y su tarea 508 de intermediario comunican entre sí mediante el mecanismo 522 de comunicación de una manera específica a las necesidades de la aplicación, operando de manera eficaz como rutinas conjuntas que comprenden una

única tarea lógica.

La Figura 6 es un diagrama de bloques esquemático de una posible configuración de hardware de sistemas de múltiples procesamientos de datos en los que puede residir al menos parcialmente la presente invención, que ilustra varias rutas potenciales para el mecanismo 522 de comunicación en la Figura 5. El primer sistema 500-1 de procesamiento de datos y el segundo sistema 500-2 de procesamiento de datos pueden incluir respectivamente el primer procesador o procesadores 602 y el segundo procesador o procesadores 604, la primera memoria 606 y la segunda memoria 608, la primera interfaz 610 de red y la segunda interfaz 612 de red, primera interfaz 614 de dispositivo de E/S y segunda interfaz 616 de dispositivo de E/S, conectadas a través del primer bus 618 y el segundo bus 620.

La ruta de comunicación más remota, de latencia más alta, pero sin embargo útil sería mediante almacenamiento 622 compartido que soporta conexiones desde tanto la primera interfaz 614 de dispositivo de E/S como la segunda interfaz 616 de dispositivo de E/S. Existe tecnología mediante la cual el mismo dispositivo de almacenamiento puede soportar la interfaz de E/S de diferentes protocolos de arquitecturas de sistemas, permitiendo de esta manera que el primer sistema 500-1 de procesamiento de datos acceda a los datos de almacenamiento del segundo sistema 500-2 de procesamiento de datos y vice-versa.

Una ruta de comunicación de latencia inferior menos remota sería mediante la red 624, que soporta conexiones desde tanto la primera interfaz 610 de red como la segunda interfaz 612 de red. Algunos protocolos de red tales como TCP/IP permiten el intercambio de paquetes de mensaje de información entre sistemas. Otros protocolos tales como Infiniband soportan VIA (Arquitectura de Interfaz Virtual) que permite compartición directa de memoria entre la primera tarea o tareas 502 y la segunda tarea o tareas 504, usando RDMA (Acceso a Memoria Directo Remoto) mediante la red 624 para permitir que el primer sistema 500-1 de procesamiento de datos acceda a la segunda memoria 608 y que el segundo sistema 500-2 de procesamiento de datos acceda a la primera memoria 606.

La ruta de comunicación de la latencia más baja menos remota implica la compartición real de memoria entre el primer y segundo sistemas 500 de procesamiento de datos, como se ilustra por el solapamiento 626 de memoria compartida entre la primera memoria 606 y la segunda memoria 608. Este tipo de compartición de memoria requiere que el primer y segundo sistemas 500 de procesamiento de datos sean particiones lógicas o físicas dentro del mismo sistema de procesamiento de datos físico. Los mismos protocolos de comunicación como se usan en la red 624 pueden usarse a la velocidad de memoria mediante el solapamiento 626 de memoria compartida, incluyendo TCP/IP e Infiniband. El último

protocolo puede ser particularmente bien adecuado a las necesidades de la presente invención, puesto que permite que la tarea 506 de descarga y la tarea 508 de intermediario interoperen como si fueran de hecho rutinas conjuntas que se ejecutan fuera de la misma memoria en el mismo sistema de procesamiento de datos físico, acercándose por lo tanto a la eficacia de los mecanismos de descarga de fin especial de la técnica anterior.

Existe tecnología y ahora está emergiendo que permite que el primer soporte lógico inalterable 514 y al segundo soporte lógico inalterable 512 de arquitecturas diferentes (por ejemplo PCMLware de Platform Solutions, Inc.) así como el primer procesador o procesadores 602 y el segundo procesador o procesadores 604 de diferentes arquitecturas (por ejemplo Interfaz de Sistema Común de Intel Corporation) coexistan dentro del mismo sistema de procesamiento de datos que puede particionarse físico. Un sistema híbrido de este tipo puede proporcionar una plataforma particularmente adecuada que posibilite la presente invención.

La Figura 7 es un diagrama de flujo que ilustra una realización de configuración de intermediario 700. El procedimiento se inicia 702 y un usuario puede seleccionar 704 una tarea nativa a descargarse, o la selección 704 puede ser implícita en alguna otra actualización a la configuración de sistema, necesitando cambios con respecto a un intermediario actual y/o tarea de descarga. Se consulta a la CMDB 706 para determinar si puede ya existir un intermediario para la tarea nativa en cuestión, y en caso afirmativo, cuál puede ser su estado con respecto al resto del sistema. Si ya existe un intermediario, se hallará en la biblioteca 708 de intermediarios. A partir de ahí puede instalarse 710 si fuera necesario 712 a través de una solicitud 714 transmitida mediante el nodo A1, activado 716 y/o reconfigurado 718 según sea necesario 720.

Si no hay intermediario correspondiente en la biblioteca 708, entonces puede buscarse 722 un equivalente remoto de la tarea nativa especificada para servir como la tarea de descarga si ya no existe ninguna. Si está disponible 724 código fuente para la tarea nativa, entonces puede compilarse de manera cruzada 726 para el sistema 500-1 de procesamiento de datos remoto. Si un equivalente remoto no existe 722 ni puede portarse, compilarse o proporcionarse automáticamente 726 de alguna manera, entonces puede solicitarse que el usuario haga una reescritura 728 de la tarea nativa o desarrolle de otra manera un equivalente remoto y lo suministre manualmente.

Una vez que se ha identificado 722 un equivalente remoto, entonces puede generarse 730 un intermediario para la tarea nativa usando una plantilla 1000 de intermediario genérica a través de una solicitud 732 transmitida mediante el nodo A1. El intermediario nuevamente generado se añade a continuación 734 a la

biblioteca de intermediario, se instala 710, y activa 716. Pueden ser necesario que se inicialicen y/o ajusten 718 otros parámetros de control para regir el despliegue y comportamiento del intermediario, como se especifica por el usuario o necesitado de manera inherente por los aspectos del entorno y configuración del sistema

5 Finalmente, la CMDB puede actualizarse 736 para reflejar la configuración modificada del sistema, que incluye el intermediario para la tarea nativa y su tarea equivalente remota asociada, ya se haya generado nuevamente o simplemente actualizado de alguna manera, y el procedimiento 700 finaliza 738.

La Figura 8 es un diagrama de flujo que ilustra una realización de generación

10 800 de intermediario. El procedimiento 800 se inicia 802 a través de una solicitud 804 recibida mediante el nodo A1. Una plantilla 1000 de intermediario genérica puede usarse 806 según se recibe 808 mediante el nodo A3, que tiene campos en blanco predeterminados para información específica relacionada con la tarea nativa a descargarse. Tales campos pueden incluir la especificación de la tarea 810 de

15 descarga equivalente remota, especificación de parámetros de entrada y salida y/o datos 812 que necesita la tarea, un alias de la tarea nativa a invocarse para fines de conmutación por recuperación 814 automática en el caso de un error, y así sucesivamente.

En una realización, la generación de intermediario puede realizarse

20 manualmente, ya sea desde cero o rellenando una plantilla de intermediario a mano. En otra realización, la generación de intermediario puede ser completamente automatizada. En una realización adicional, puede generarse automáticamente un intermediario básico y a continuación personalizarse y optimizarse a mano si se desea.

Puede también realizarse automáticamente optimización 816 de rendimiento

25 adicional. Por ejemplo, si una tarea nativa dada tiene entrada o salida implícita que no necesita especificarse, la porción o porciones relacionadas de la plantilla de intermediario pueden eliminarse automáticamente del intermediario según se generan. Pueden emplearse otras estrategias de este tipo por el extremo trasero de

30 un compilador de optimización en la generación de intermediario, y el procedimiento 800 finaliza 818.

La Figura 9 es un diagrama de flujo que ilustra una realización de instalación

900 de intermediario, después de que se haya generado un intermediario 730 o recuperado desde la biblioteca 708 de intermediarios. El procedimiento 900 se inicia

35 902 a través de una solicitud 904 recibida mediante el nodo A2. En la preparación para la eventualidad de un error de condición de descarga, el alias de conmutación por recuperación puede señalarse 906 a la tarea nativa. Este alias se usa para invocar la tarea nativa directamente en el caso de un error en el proceso de

descarga. Sin eso, un error de descarga puede provocar un bucle indefinido ya que el intermediario vuelve a invocar esencialmente de manera repetitiva, intentando insatisfactoriamente descargar la tarea nativa una y otra vez. El alias de conmutación por recuperación podría también usarse para omitir de manera deliberada el intermediario cuando no se desea la descarga de la tarea nativa en una instancia específica, sin tener que desactivar el intermediario en total.

Cualesquiera parámetros de control que sean necesarios deberían a continuación establecerse 908 según sean apropiados. Por ejemplo, tales parámetros pueden incluir información de conectividad entre el sistema nativo y sistema 500-1 de procesamiento de datos remoto, tal como nombres del Servidor de Nombres de Dominio (“DNS”) o direcciones del Protocolo de Internet (“IP”). Los parámetros pueden incluir filtros que posibilitan o no posibilitan la descarga cuando se cumplen criterios específicos, tales como coincidencia de todo o parte de un nombre de trabajo con uno o más patrones de carácter. Los parámetros pueden incluir la habilitación o deshabilitación de mecanismos de transporte de datos específicos, tales como el Protocolo de Transferencia de Ficheros (“FTP”), Movedor de Datos Paralelo (“PDM”), y así sucesivamente. Puede proporcionarse un parámetro para habilitar o deshabilitar la descarga de la tarea nativa. En una realización, puede proporcionarse un conjunto de datos de la biblioteca de parámetros z/OS® (“parmlib”) con miembros que contienen los parámetros de control para cada intermediario. En una realización adicional, puede proporcionarse un “panel de control” u otra interfaz de usuario integrada para inspeccionar y establecer los diversos parámetros.

Finalmente, puede señalarse 910 un alias de la tarea nativa al intermediario, para invocar de manera transparente la tarea de descarga en lugar de la tarea nativa. Pueden emplearse también otros procedimientos similares, tales como salidas especificadas por usuario para modificar la invocación de tarea al vuelo. Para instalar un intermediario, las etapas anteriores pueden simplemente deshacerse en orden inverso 910, 908 y 906, y el procedimiento 900 finaliza 912.

La Figura 10 es un diagrama de flujo que ilustra una realización de una plantilla 1000 de intermediario que puede transmitirse 1002 mediante el nodo A3. El procedimiento 1000 se inicia 1004 y pueden leerse 1006 parámetros de control desde una parmlib u otra fuente. Si el intermediario no está activo o habilitado 1008, entonces todo el proceso de descarga se omite 1010, invocando la tarea nativa a través del alias de conmutación por recuperación. No se solicita conmutación por recuperación en este caso, puesto que la desactivación o deshabilitación del intermediario no es un error en y de sí mismo, aunque puede o puede no haber resultado de un error anterior de alguna clase.

Si el intermediario está activo 1008, entonces se realiza la descarga. Una solicitud de descarga puede componerse 1012, que comprende los campos que se rellenaron por la generación 800 de intermediario, tal como la tarea 810 de descarga equivalente remota, parámetros de entrada y/o salida y/o datos 812, y así sucesivamente. La solicitud de descarga puede a continuación enviarse 1014 al agente de descarga remoto según se transmite 1016 mediante el nodo A4. En una realización, el intermediario puede interrogar 1018 el estado de la solicitud de descarga remota en un bucle de espera de interrogación. En otra realización, puede esperar 1018 una respuesta explícita mediante una interrupción desde el agente de descarga. En una realización adicional, puede utilizarse una combinación 1018 de interrogación e interrupción.

Después de esperar 1018, el intermediario comprueba 1020 el estado de la solicitud de descarga. Si tuviera lugar algún error 1022 o existiera 1024 condición de bloqueo persistente que evitara la finalización satisfactoria de la solicitud de descarga, se invoca la tarea nativa 1026 a través del proceso de conmutación por recuperación, y se solicita 1028 conmutación por recuperación según se transmite 1030 mediante el nodo A5. En una realización, la condición 1024 de bloqueo puede detectarse por ausencia de progreso hacia adelante visible por el agente de descarga después de uno o más intervalos 1018 de interrogación, tal como puede indicarse por una indicación de tiempo que falla a avanzar como parte de una comprobación de "mantenimiento de la conexión".

Si el sistema 500-2 de procesamiento de datos nativo cancela 1032 la tarea nativa que se está descargando por el intermediario, entonces el intermediario a su vez debe cancelar 1034 la tarea equivalente remota a través de una solicitud según se transmite 1036 mediante el nodo A6. Si el estado indica que la tarea equivalente remota se ha completado 1038 satisfactoriamente, entonces pueden realizarse 1040 cualesquiera acciones de recapitulación necesarias, y el procedimiento 1000 de intermediario se completa 1042. Las acciones 1040 de recapitulación pueden incluir desasignar recursos, informar estado al sistema 500-2 de procesamiento de datos nativo, y así sucesivamente. Si no existe ninguna de las condiciones anteriores, entonces el intermediario reanuda la espera 1018.

La descripción anterior incluye los elementos básicos de un intermediario sencillo. En una realización adicional, pueden proporcionarse otros elementos condicionales en la plantilla 1000 para soportar tareas más avanzadas, para eliminarse en casos más sencillos que no requieren estos elementos. Por ejemplo, una tarea nativa más avanzada puede comprender un subsistema mayor que necesita que se lance inicialmente o se active de otra manera, a continuación de servicio a solicitudes en tiempo real de otras tareas y/o usuarios, y finalmente se

apague o se desactive de otra manera cuando ya no sea necesaria. En una realización, el subsistema nativo puede descargarse mediante un único intermediario complejo. En otra realización, puede descomponerse en múltiples intermediarios más sencillos.

5           En general, puede ser deseable mantener el módulo 202 de plantilla de intermediario tan sencillo como sea posible, para minimizar el tiempo de CPU nativo, en la medida en que la descarga el sistema 500-2 de procesamiento de datos nativo es un objetivo establecido de acoplamiento de intermediario. Es decir, también porque el módulo 204 de generación de intermediario puede preprogramar la  
10 información específica relacionada con la tarea nativa a descargarse rellenando campos en blanco predeterminados en la plantilla de intermediario, a diferencia de tener el intermediario 508 que incurrir en la sobrecarga de determinación de esa información en tiempo real. Las optimizaciones de rendimiento realizadas por el módulo 204 de generación de intermediario como se ha descrito anteriormente  
15 pueden también ayudar a servir al mismo fin, en concreto, a minimizar adicionalmente el tiempo de CPU nativo.

La Figura 11 es un diagrama de flujo que ilustra una realización de un agente 1100 de descarga. Véase la Figura 11A. El procedimiento 1100A se inicia 1102A a través de una solicitud según se transmite 1104A mediante el nodo A10. Mientras  
20 está en reposo, el agente de descarga simplemente espera 1106A para nueva solicitud o solicitudes de descarga 1108A según se reciben 1110A mediante el nodo A4 desde uno o más intermediarios. Como con el intermediario, el agente de descarga puede utilizar interrogación y/o interrupción. Si el intermediario y el agente de descarga ambos usan interrogación, entonces cualquier diferencia en frecuencia  
25 de interrogación puede factorizarse en detección de bloqueo como parte de una comprobación de mantenimiento de la conexión, tal como esperar más de un intervalo de interrogación para determinar progreso hacia delante. Pueden proporcionarse colas para que el agente de descarga rastree el progreso de las tareas de descarga equivalentes remotas, que incluyen una cola pendiente de solicitudes de descarga que se han recibido y una cola activa de tareas de descarga  
30 que están en progreso.

Cuando se recibe una nueva solicitud de descarga, entonces puede añadirse 1112A a la cola pendiente. Si se cancela 1114A una o más solicitudes de descarga a través de una solicitud recibida 1116A mediante el nodo A6, entonces cualesquiera  
35 tareas de descarga correspondientes pueden terminarse 1118A y las correspondientes solicitudes de descarga pueden eliminarse 1120A de la cola pendiente y la cola activa. El procedimiento 1100A finaliza 1022A y continúa 1124A al procedimiento 1100B mediante el nodo A7.

Véase la Figura 11B. El procedimiento 1100B se inicia 1102B a través de una solicitud según se recibe 1104B mediante el nodo A7. Si la cola activa no está vacía 1106B, a continuación puede procesarse 1108B una entrada consecutiva a la vez. La tarea de descarga asociada con la siguiente entrada puede comprobarse 1110B, y su estado actualizarse 1112B por consiguiente, para informarse al intermediario correspondiente. También puede incluirse 1114B una indicación de tiempo en el estado como parte del mecanismo de detección de bloqueo de mantenimiento de la conexión anteriormente descrito.

Si la tarea se finaliza 1116B, entonces puede eliminarse 1118B la entrada correspondiente de la cola activa. El tiempo de la CPU (Unidad de Procesamiento Central) gastado para la tarea puede añadirse 1120B a un tiempo de CPU total, para mantener la pista de cuánto tiempo de CPU se está gastando en tareas de descarga. Esta información de temporización puede ser necesaria para licencia basada en uso de la instalación de descarga de acoplamiento de intermediario.

En una realización adicional, si se solicita 1122B supervisión de sistema, entonces los datos de medición solicitados pueden acumularse 1124B, y crearse 1126B uno o más registros para mantener los datos. También puede incluirse 1128B una indicación de tiempo en el registro o registros. El registro o registros pueden grabarse 1130B y el supervisor de sistema de procesamiento de datos nativo informarse 1132B de que los datos solicitados están disponibles para inspección. El procedimiento 1100B finaliza 1134B y continúa 1136B al procedimiento 1100C mediante el nodo A8.

Véase la Figura 11C. El procedimiento 1100C se inicia 1102C a través de una solicitud según se recibe 1104C mediante el nodo A8. Como ya se ha indicado, la licencia de la instalación de acoplamiento de intermediario puede estar basada en uso. En una realización, la licencia puede ser para un número especificado de Millones de Instrucciones por Segundo (“MIPS”). El sistema de descarga remoto puede tener una capacidad de MIPS total, de la cual el número con licencia de MIPS es una fracción que puede calcularse dividiéndola en los MIPS totales. El uso puede calcularse a continuación a través de intervalos de medición regulares.

Si el tiempo de CPU total según se ha calculado anteriormente es menor que el tiempo total que ha transcurrido durante el intervalo de medición actual, temporiza la fracción con licencia de la capacidad de MIPS total, entonces la capacidad de descarga adicional está aún disponible 1106C.

Una vez que el tiempo total transcurrido alcanza la cantidad del intervalo 1108C de medición especificado, entonces el tiempo total 1110C y el tiempo de CPU total 1112C pueden resetearse a cero, iniciando por lo tanto el siguiente intervalo de medición. Cuanto más largo sea el intervalo de medición, más capacidad de

descarga sin uso puede “cubrirse” para su uso futuro. Puede ser aconsejable hacer el intervalo de medición lo suficientemente largo para una cierta cantidad de capacidad de cubrición para manejar picos en demanda de descarga, pero lo suficientemente corto para racionar de manera equitativa la capacidad con licencia a través del tiempo de una manera fija predecible. Las realizaciones pueden incluir un intervalo de medición de una hora, cuatro horas, ocho horas, veinticuatro horas, o similares.

Si la capacidad de descarga adicional está disponible 1108C, y la cola pendiente no está vacía 1114C, entonces puede procesarse 1116C una entrada consecutiva cada vez. La tarea de descarga asociada con la siguiente entrada puede lanzarse 1118C o activarse de otra manera, eliminarse 1120C de la cola pendiente, y añadirse 1122C a la cola activa. Si no está disponible 1106C capacidad adicional o todas las tareas de descarga pendientes se han lanzado 1118C y la cola pendiente está ahora vacía 1114C, entonces el agente de descarga remoto continúa la operación 1124C mediante el nodo A9 a menos que se detenga 1126C caso en el que el procedimiento 1100C finaliza 1128C.

Véase la Figura 11D. El procedimiento 1100D se inicia 1102D a través de una solicitud según se recibe 1104D mediante el nodo A9. En una realización, el agente de descarga remoto puede comprobar la validez de una clave de licencia mientras se entra en el estado en reposo o en cualquier otra coyuntura apropiada. Puede usarse un patrón de clave o un patrón de datos como entrada a un algoritmo de seguridad de capacidad para generar inicialmente la clave de licencia basándose en información válida conocida, y a continuación realizar la comprobación de validez anteriormente mencionada regenerando la clave basándose en información actual, y adaptándola contra la clave que se generó inicialmente.

El patrón de clave puede ser un valor que es único a un entorno de sistema con licencia, tal como una dirección de IP de una interfaz de Ethernet. El patrón de datos puede ser una cadena de caracteres que comprende una fecha de inicio (DDMMYYYY) desde cuándo ha de tener efecto la licencia, un día de inicio del año (NNN), un día actual del año (NNN), un número de días (NNN), un recuento de capacidad con licencia (NN), y así sucesivamente. En una realización, el recuento de capacidad con licencia puede ser un porcentaje de capacidad de MIPS total, que puede corresponder también a un porcentaje de tiempo transcurrido que comprende tiempo de CPU consumido por procesamiento de descarga durante un intervalo de medición, como se ha descrito anteriormente.

El patrón de clave puede convertirse 1106D a un número entero de tal manera que ambas operaciones aritméticas y lógicas pueden realizarse en él mismo. En una realización, el número entero puede ser binario. En una realización

adicional, la base del número entero puede ser una potencia de dos superior, tal como un cuarteto de cuatro bits que comprende un dígito hexadecimal.

Las constantes de cadena numérica, también denominadas como números “mágicos”, pueden predeterminarse y usarse para operar en el número entero. El número entero puede multiplicarse 1108D por un primer número mágico. En una 5 realización, la multiplicación 1108D puede realizarse en uno o más dígitos a la vez, incluyendo bits, cuartetos, bytes, y así sucesivamente. Un segundo número mágico, que puede o puede no ser el mismo que el primer número mágico, puede añadirse 1110D al número entero. En el número entero puede también realizarse la operación 10 OR 1112D con una versión desplazada 1114D de sí mismo. Por ejemplo, puede desplazarse a la izquierda en uno o más lugares, incluyendo un bit, cuarteto, byte, y así sucesivamente.

En el número entero puede realizarse también la operación OR 1116D con el patrón de datos. Si el patrón de datos es más largo que el número entero, de 15 manera que hay más datos restantes 1118D, entonces en la porción 1120D restante puede realizarse la operación OR 1116D con el resultado de la operación OR 1116D anterior hasta que se agoten los datos 1118D. En una realización, en el número entero resultante de las operaciones tras el patrón de clave puede realizarse la operación OR con el patrón de datos de una manera por fragmentos. En una 20 realización adicional, puede realizarse la operación OR 1116D de izquierda a derecha un bit, cuarteto, o byte a la vez. Por ejemplo, en el primer byte del patrón de clave resultante puede realizarse la operación OR 1116D con el primer byte del patrón de datos, en el segundo byte del patrón de clave puede a continuación realizarse la operación OR 1116D con el segundo byte del patrón de datos, y así 25 sucesivamente. Cuando el patrón de clave se agota y hay aún 1118D datos en el patrón de datos a continuación el patrón de clave puede resetearse al primer byte y puede reanudarse la operación OR del patrón de datos 1116D. Esto puede continuar hasta que se agote 1118D el patrón de datos.

Los tipos de operaciones aritméticas y lógicas realizadas, y la secuencia en la 30 que se realizan, no están limitadas a aquellas en la descripción anterior. Una vez que el procesamiento del patrón de clave y el patrón de datos está completo, entonces el valor clave resultante se compara 1122D con el valor de clave inicialmente generado. Si coinciden, entonces la operación del agente de descarga remoto continúa y puede volver 1124D al estado en espera mediante el nodo A10 y 35 esperar 1106A durante algún tiempo más para pasare. Si no coinciden 1122D, entonces se suspende la operación y el procedimiento 1100D finaliza 1126D. Otros módulos puede verse también directa o indirectamente afectados por la suspensión de la operación. Por ejemplo, la suspensión del agente de descarga remoto puede

provocar que el intermediario detecte un bloqueo, que puede a su vez provocar que se solicite conmutación por recuperación .

La Figura 12 es un diagrama de flujo que ilustra una realización de conmutación por recuperación 1200 . El procedimiento 1200 se inicia 1202 y si se solicita conmutación por recuperación 1204 según se recibe 1206 mediante el nodo A5, entonces la instalación de automatización de operaciones nativas puede estar informada 1208. Puede realizarse una determinación 1210 de la cantidad de capacidad de descarga que se ha perdido. La capacidad nativa puede mejorarse dinámicamente 1212 para componer temporalmente la capacidad de descarga perdida. En una realización, puede utilizarse la instalación de Mejora de Capacidad bajo Demanda (“CUoD”). Cualesquiera intermediarios afectados pueden a continuación desactivarse 1214. En una realización, los parámetros de control de activación específicos de intermediario pueden desactivarse. En otra realización, puede desactivarse un parámetro de activación de acoplamiento de intermediario a nivel de sistema.

Cuando se ha resuelto 1216 la interrupción, los intermediarios afectados pueden reactivarse 1218, permitiendo que se reanude la descarga. La capacidad nativa que se mejoró temporalmente puede degradarse 1220 dinámicamente, restaurando las operaciones de acoplamiento de intermediario normales, y el procedimiento 1200 finaliza 1222.

Las realizaciones pueden ponerse en práctica en otras formas específicas. Las realizaciones descritas se han de considerar en todos los aspectos únicamente como ilustrativas y no restrictivas. El alcance de la invención se indica, por lo tanto, por las reivindicaciones adjuntas en lugar de por la descripción anterior. Todos los cambios que entran dentro del significado y alcance de equivalencia de las reivindicaciones se han de abarcar dentro de su alcance.

## **REIVINDICACIONES**

1. Un aparato que comprende:
  - un módulo de plantilla de intermediario que está configurado para descargar  
5 de manera transparente una tarea de un sistema de procesamiento de datos nativo a una tarea equivalente en un sistema de procesamiento de datos remoto;
  - un módulo de generación de intermediario que rellena la plantilla de intermediario con información para una tarea nativa seleccionada para generar un intermediario de la tarea nativa, en el que la información comprende una  
10 especificación de una tarea de descarga que es un equivalente remoto de la tarea nativa; y
  - un módulo de agente de descarga que recibe una solicitud desde el intermediario para realizar la tarea de descarga en el sistema de procesamiento de datos remoto
  - 15 en el que el módulo de plantilla de intermediario, el módulo de generación de intermediario, y el módulo de agente de descarga comprenden uno o más de hardware lógico y código ejecutable, el código ejecutable almacenado en uno o más medios legibles por ordenador no transitorios.
- 20 2. El aparato de la reivindicación 1, en el que el módulo de plantilla de intermediario está configurado para supervisar el agente de descarga, cancelar la solicitud para el agente de descarga, e iniciar una conmutación por recuperación para revertir a la tarea nativa.
- 25 3. El aparato de la reivindicación 1, en el que la información rellena por el módulo de generación de intermediario comprende adicionalmente especificación de datos de entrada y salida que necesita la tarea nativa.
4. El aparato de la reivindicación 1, en el que la información rellena por el  
30 módulo de generación de intermediario comprende adicionalmente especificación de un alias de la tarea nativa a invocarse para los fines de conmutación por recuperación automática en el caso de un error.
5. El aparato de la reivindicación 1, en el que el módulo de agente de descarga  
35 pone en cola una solicitud de descarga desde el intermediario a una cola pendiente.
6. El aparato de la reivindicación 1, en el que el módulo de agente de descarga mueve la solicitud desde una cola pendiente a una cola activa y activa la

correspondiente tarea de descarga.

- 5 7. El aparato de la reivindicación 1, en el que el módulo de agente de descarga supervisa la tarea de descarga en una cola activa y quita de la cola la tarea de descarga tras la finalización.
- 10 8. El aparato de la reivindicación 1, que comprende adicionalmente un módulo de configuración de intermediario que incorpora el intermediario y su correspondiente tarea de descarga en una base de datos de gestión de configuración.
- 15 9. El aparato de la reivindicación 8, en el que el módulo de configuración de intermediario mejora la seguridad y control de acceso para código y datos de intermediario.
- 20 10. El aparato de la reivindicación 8, en el que el módulo de configuración de intermediario ajusta parámetros de control del intermediario y activa inicialmente el intermediario.
- 25 11. El aparato de la reivindicación 1, que comprende adicionalmente un módulo de instalación de intermediario que instala el intermediario generado en el sistema de procesamiento de datos nativo.
- 30 12. El aparato de la reivindicación 11, en el que el módulo de instalación de intermediario crea parámetros de control para el intermediario.
- 35 13. El aparato de la reivindicación 11, en el que el módulo de instalación de intermediario asocia un alias de tarea nativa con el intermediario.
14. El aparato de la reivindicación 1, que comprende adicionalmente un módulo de conmutación por recuperación que revierte de vuelta a la tarea nativa si el intermediario, módulo de agente de descarga, o sistema de procesamiento de datos remoto experimentan una interrupción.
15. El aparato de la reivindicación 14, en el que el módulo de conmutación por recuperación ajusta la capacidad del sistema de procesamiento de datos nativo, desactiva cualquier intermediario afectado, y restaura la operación de acoplamiento de intermediario normal cuando se ha resuelto la interrupción.

16. Un sistema que comprende:
- un sistema de procesamiento de datos remoto comprendido de un primer sistema operativo, un primer soporte lógico inalterable, y una primera plataforma de hardware;
  - un sistema de procesamiento de datos nativo comprendido de un segundo sistema operativo, un segundo soporte lógico inalterable, y una segunda plataforma de hardware;
  - un mecanismo de comunicación entre el sistema de procesamiento de datos remoto y el sistema de procesamiento de datos nativo;
  - un módulo de plantilla de intermediario que está configurado para descargar de manera transparente una tarea del sistema de procesamiento de datos nativo a una tarea equivalente en el sistema de procesamiento de datos remoto;
  - un módulo de generación de intermediario que rellena la plantilla de intermediario con información para una tarea nativa seleccionada para crear un intermediario de la tarea nativa, en el que la información comprende una especificación de una tarea de descarga que es un equivalente remoto de la tarea nativa;
  - un módulo de agente de descarga que recibe una solicitud desde el intermediario para realizar la tarea de descarga en el sistema de procesamiento de datos remoto en el que el módulo de plantilla de intermediario, el módulo de generación de intermediario, y el módulo de agente de descarga comprenden uno o más de hardware lógico y código ejecutable, el código ejecutable almacenado en uno o más medios legibles por ordenador no transitorios; y
  - un sistema de almacenamiento para almacenar datos de entrada y salida.
17. El sistema de la reivindicación 16, que comprende adicionalmente un módulo de conmutación por recuperación que informa a un módulo de automatización de operaciones cuando tiene lugar una conmutación por recuperación .
18. El sistema de la reivindicación 16, en el que el módulo de agente de descarga informa a un módulo de supervisión de sistema de datos de medición solicitados.
19. Un producto de programa informático que comprende un medio legible por ordenador no transitorio que tiene código de programa utilizable por ordenador ejecutable para realizar operaciones para procesamiento de datos de descarga, comprendiendo las operaciones del producto de programa informático:

proporcionar una plantilla de intermediario que está configurada para descargar de manera transparente una tarea de un sistema de procesamiento de datos nativo a una tarea equivalente en un sistema de procesamiento de datos remoto;

5 rellenar la plantilla de intermediario con información para una tarea nativa seleccionada para crear un intermediario de la tarea nativa, en el que la información comprende una especificación de una tarea de descarga que es un equivalente remoto de la tarea nativa; y

10 recibir una solicitud del intermediario para realizar la tarea de descarga en el sistema de procesamiento de datos remoto.

20. Un procedimiento implementado por máquina para procesamiento de datos de descarga, comprendiendo el procedimiento las etapas de:

15 proporcionar una plantilla de intermediario que está configurada para descargar de manera transparente una tarea de un sistema de procesamiento de datos nativo a una tarea equivalente en un sistema de procesamiento de datos remoto;

20 rellenar la plantilla de intermediario con información para una tarea nativa seleccionada para crear un intermediario de la tarea nativa, en el que la información comprende una especificación de una tarea de descarga que es un equivalente remoto de la tarea nativa; y

recibir una solicitud del intermediario para realizar la tarea de descarga en el sistema de procesamiento de datos remoto.

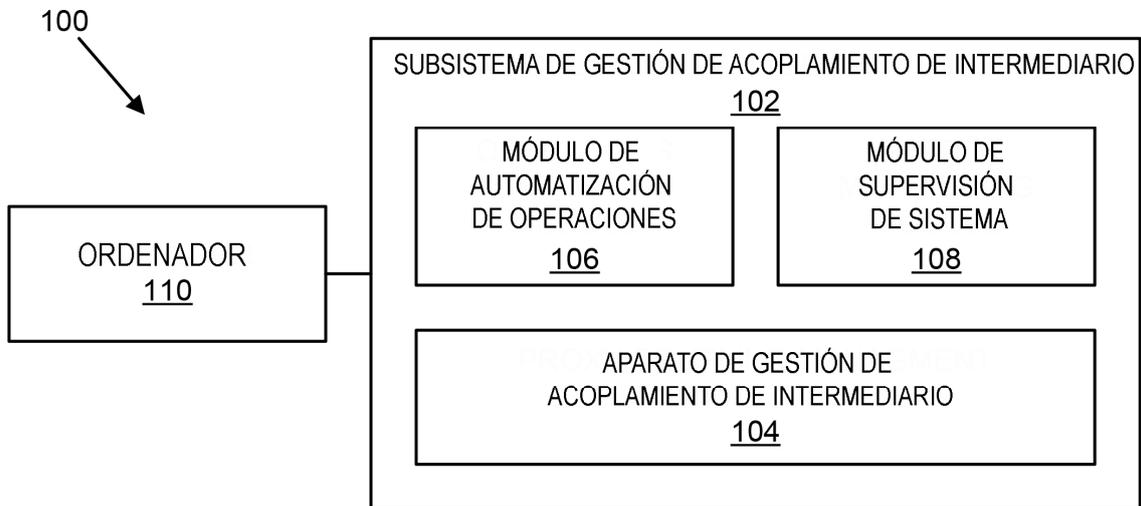


FIG. 1

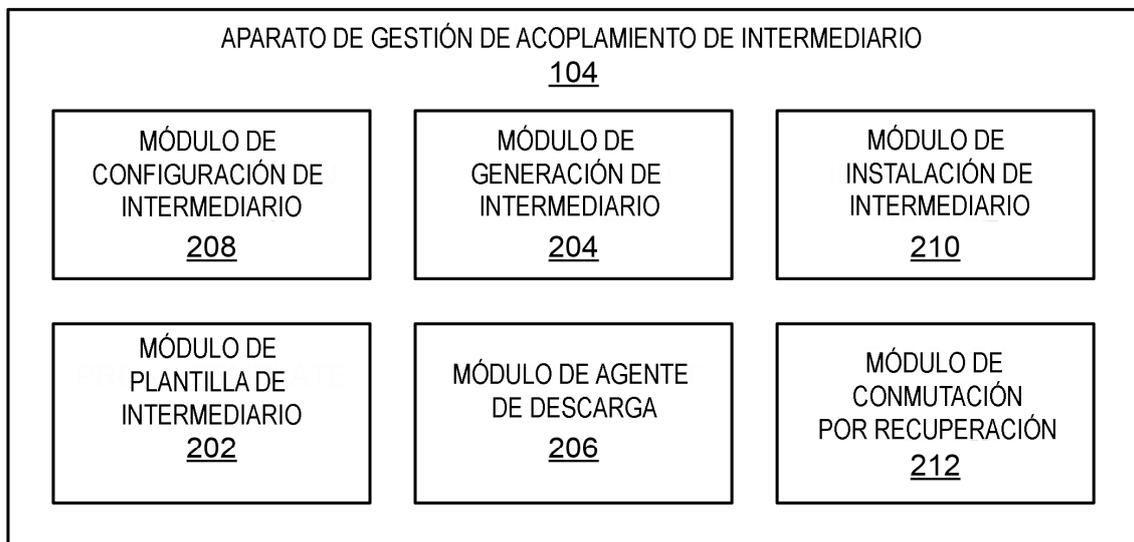


FIG. 2

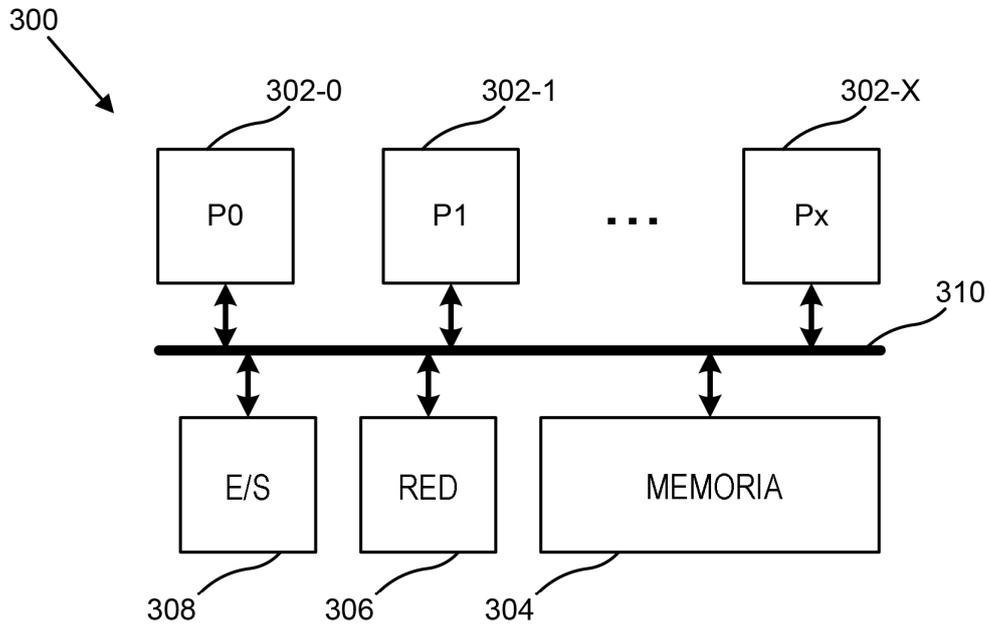


FIG. 3

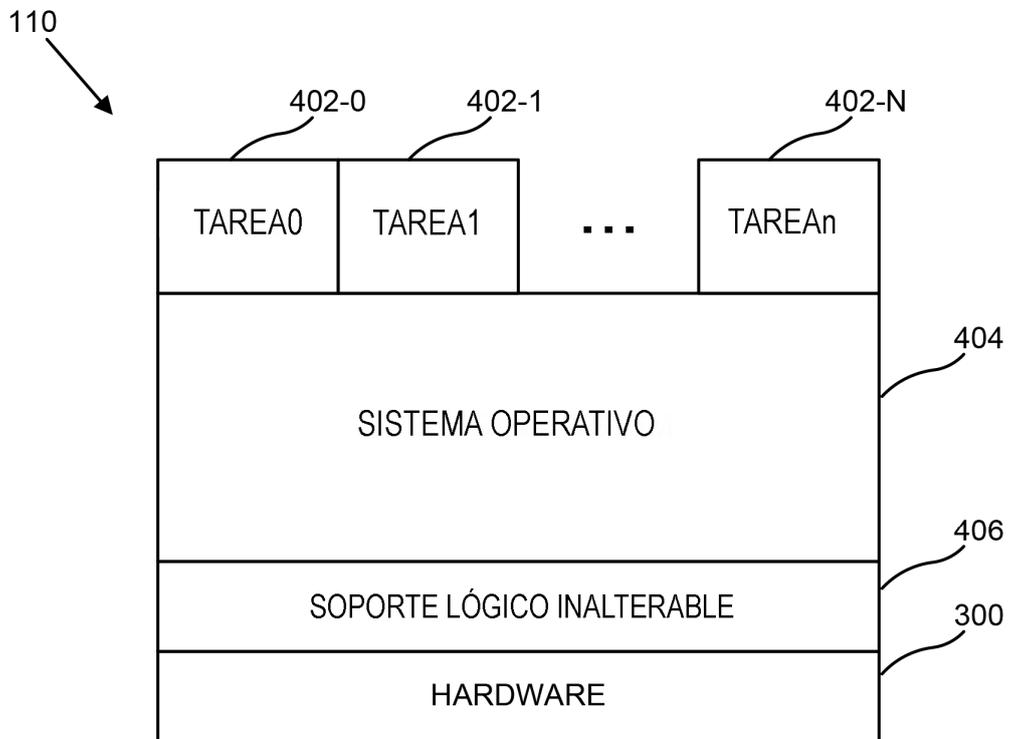


FIG. 4

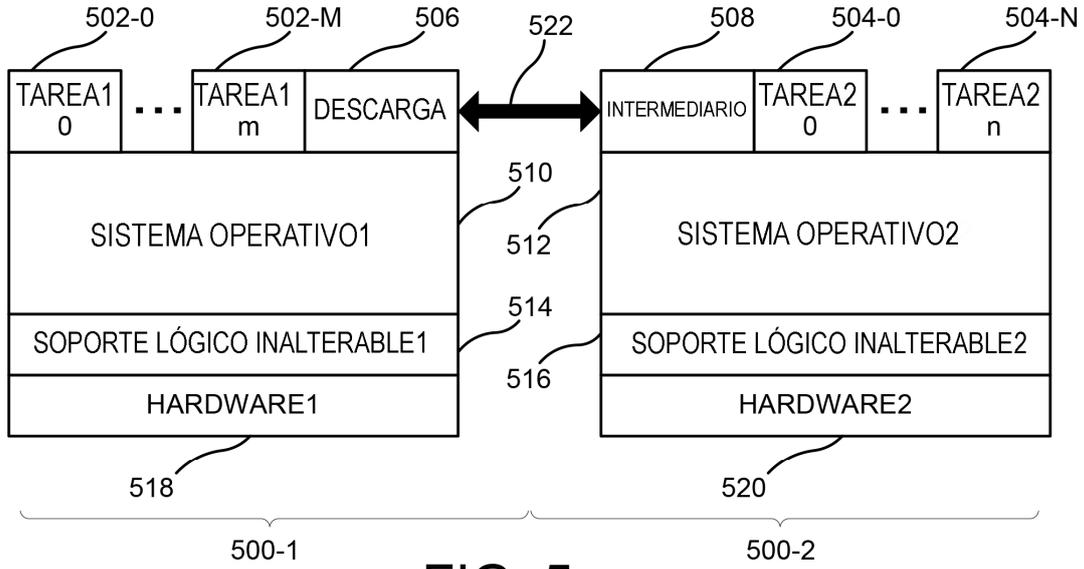


FIG. 5

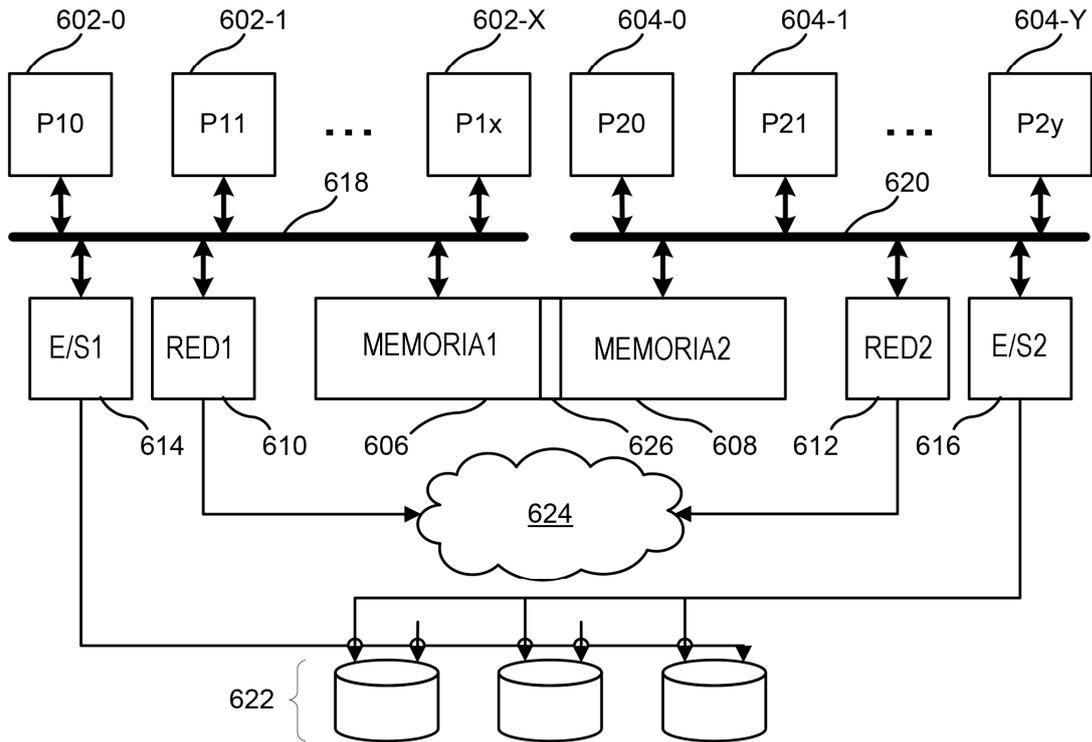


FIG. 6

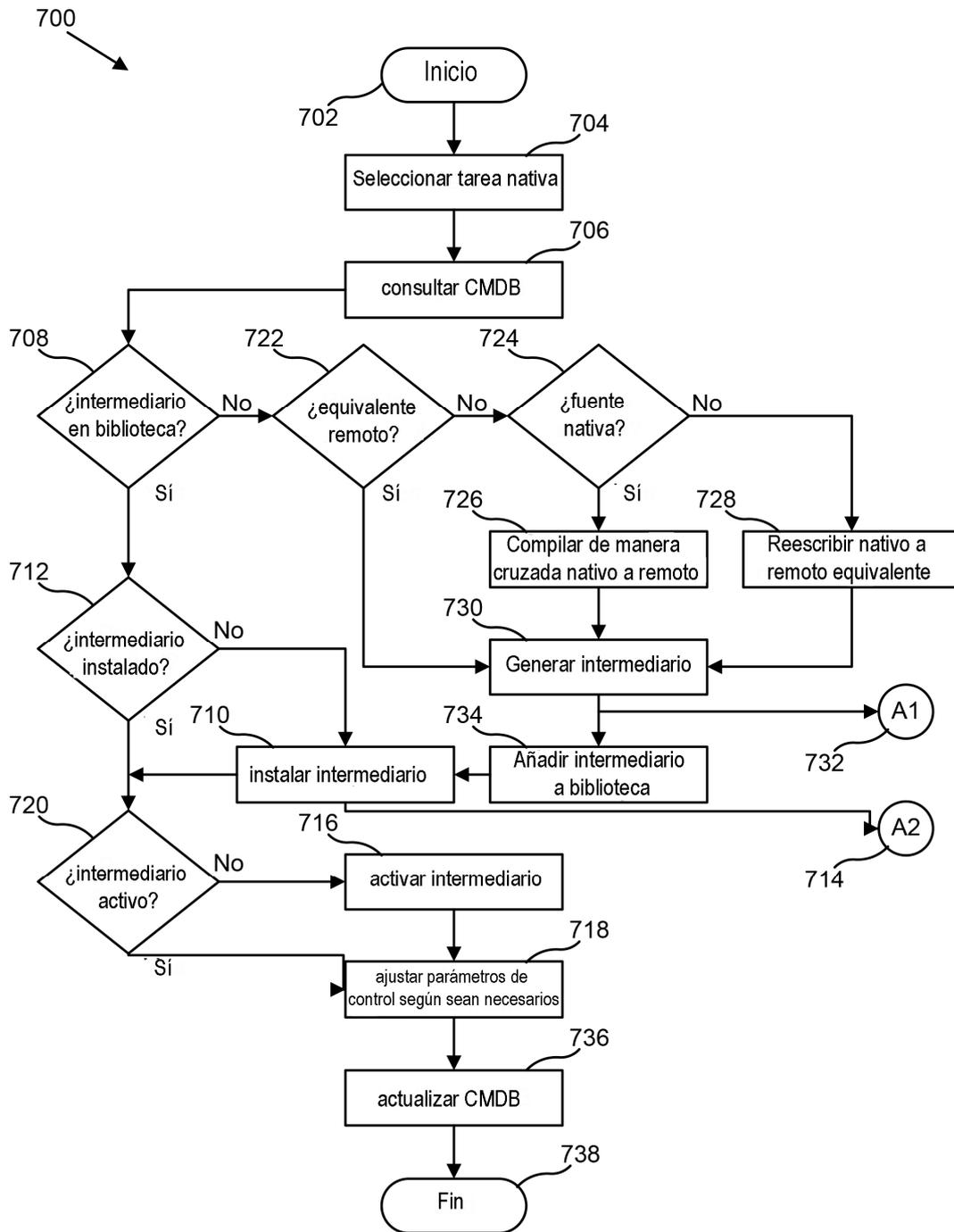


FIG. 7

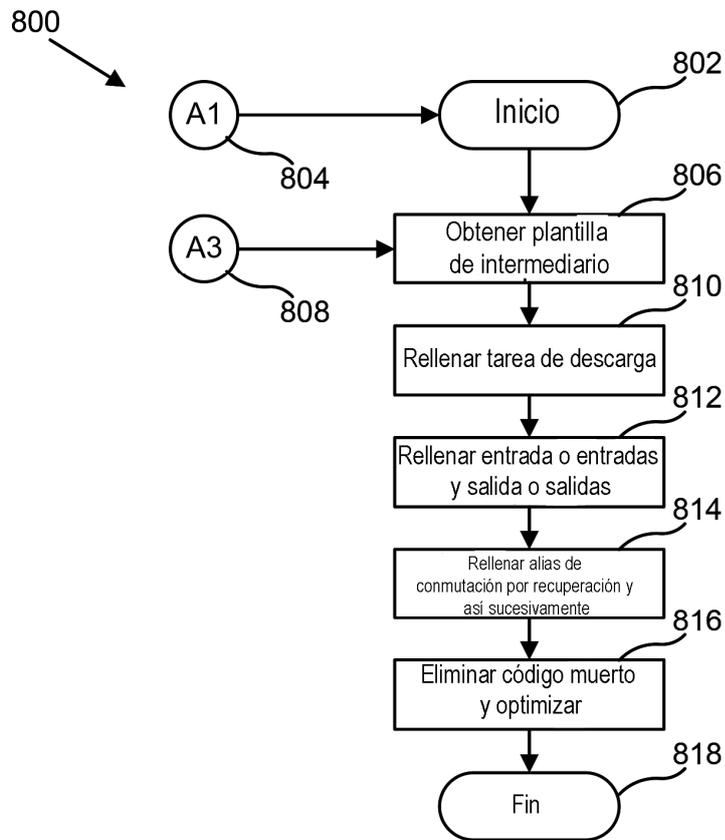


FIG. 8

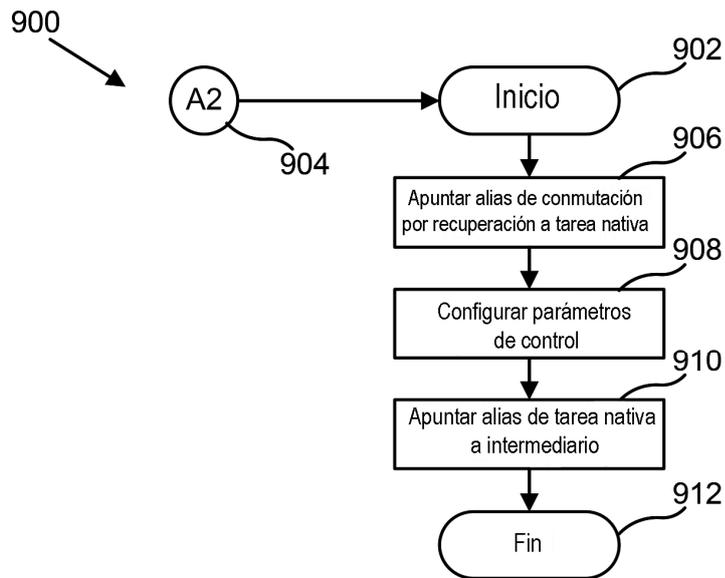


FIG. 9

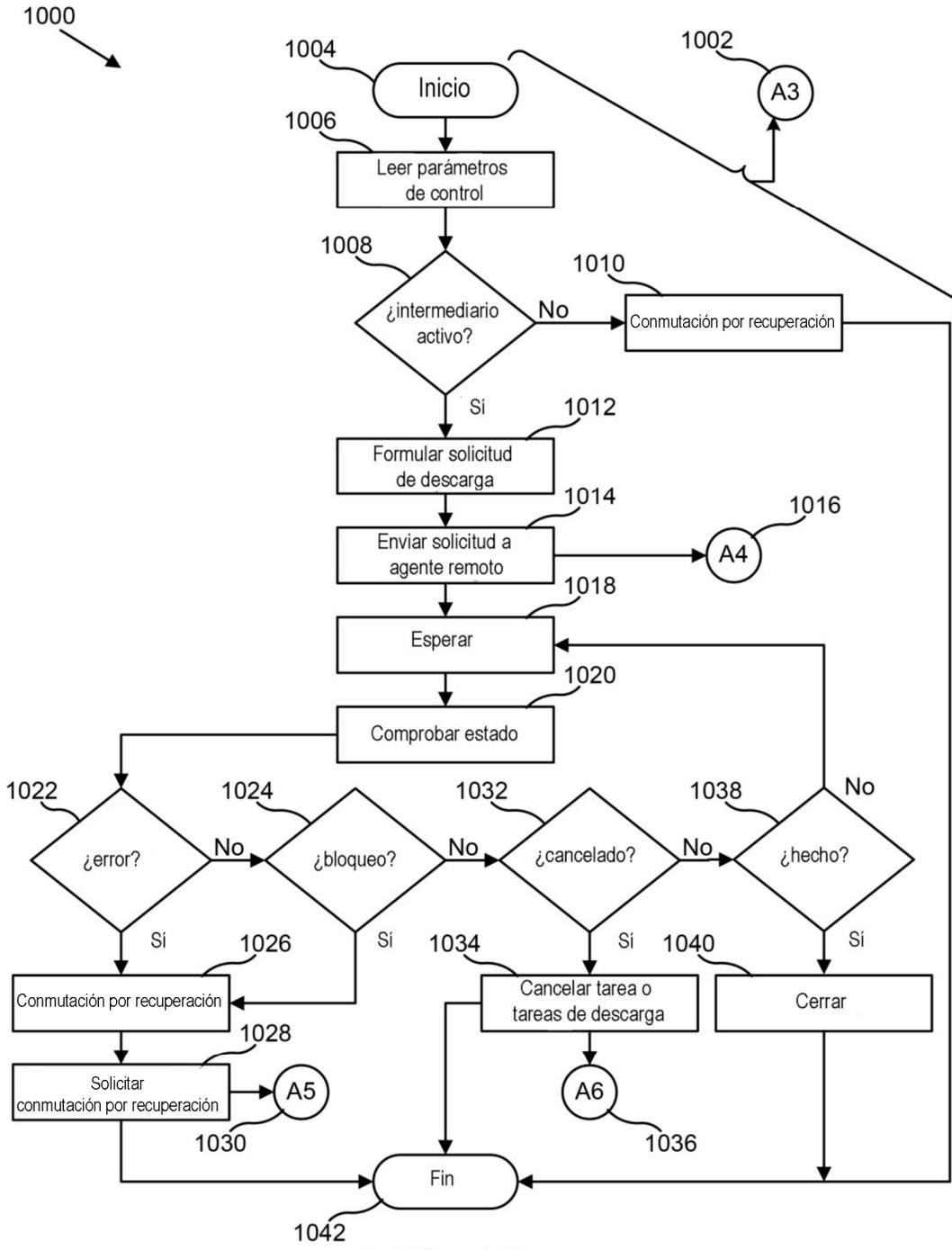


FIG. 10

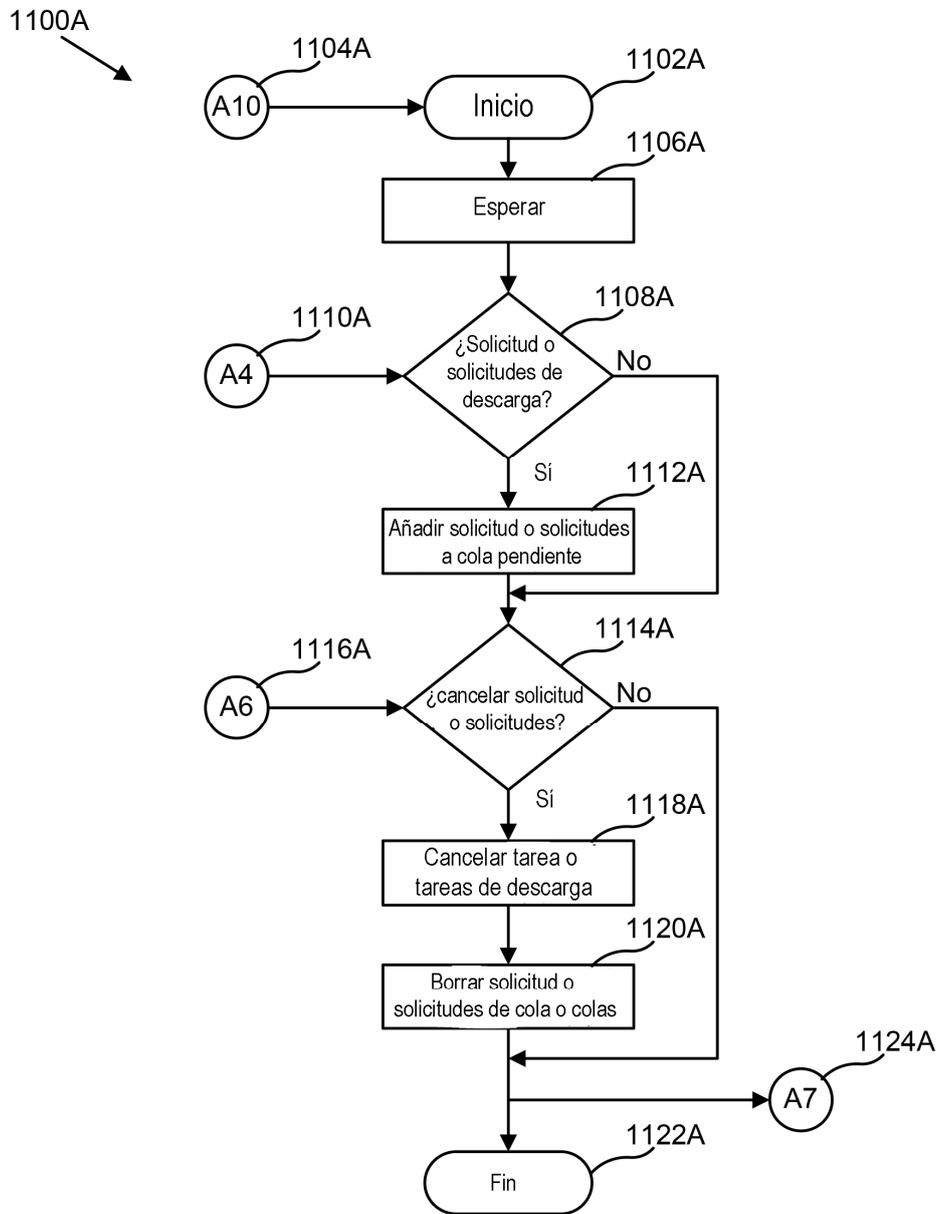


FIG. 11A

1100B

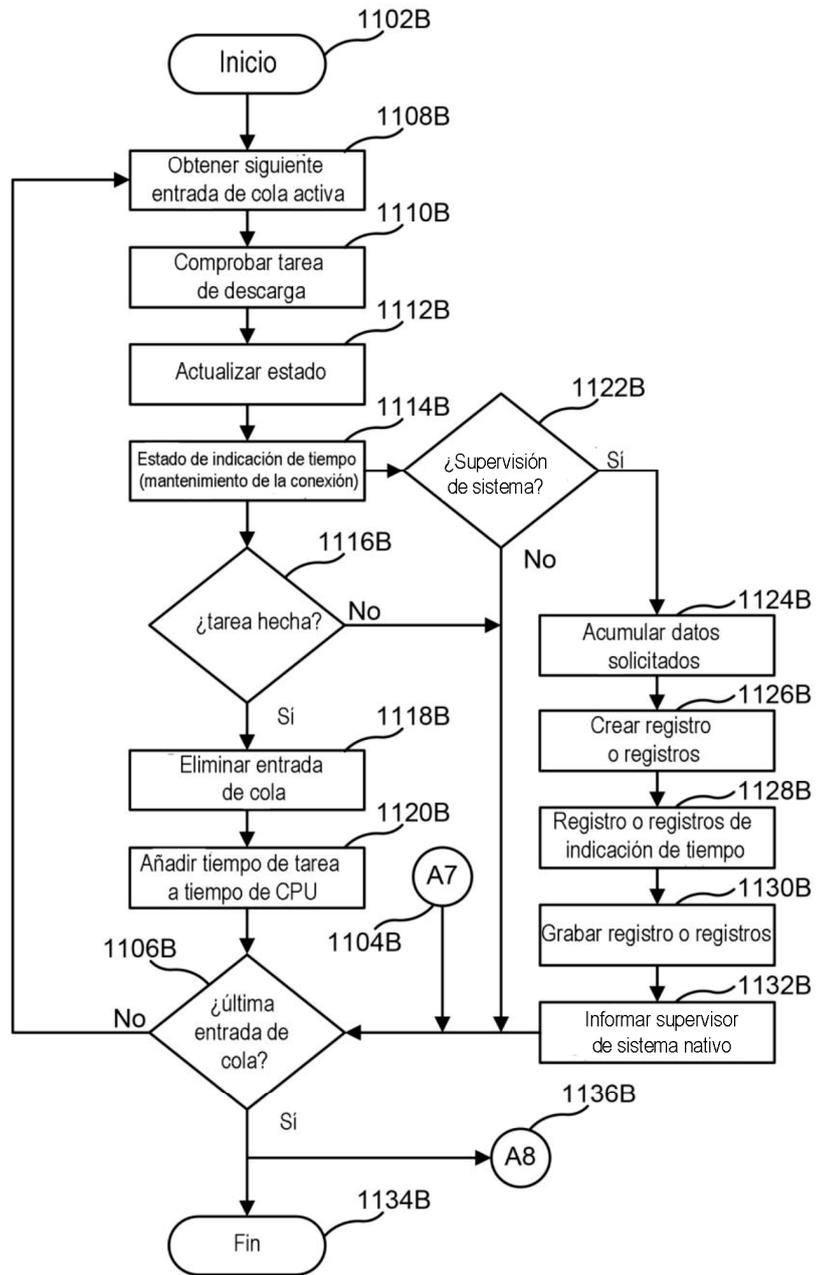


FIG. 11B

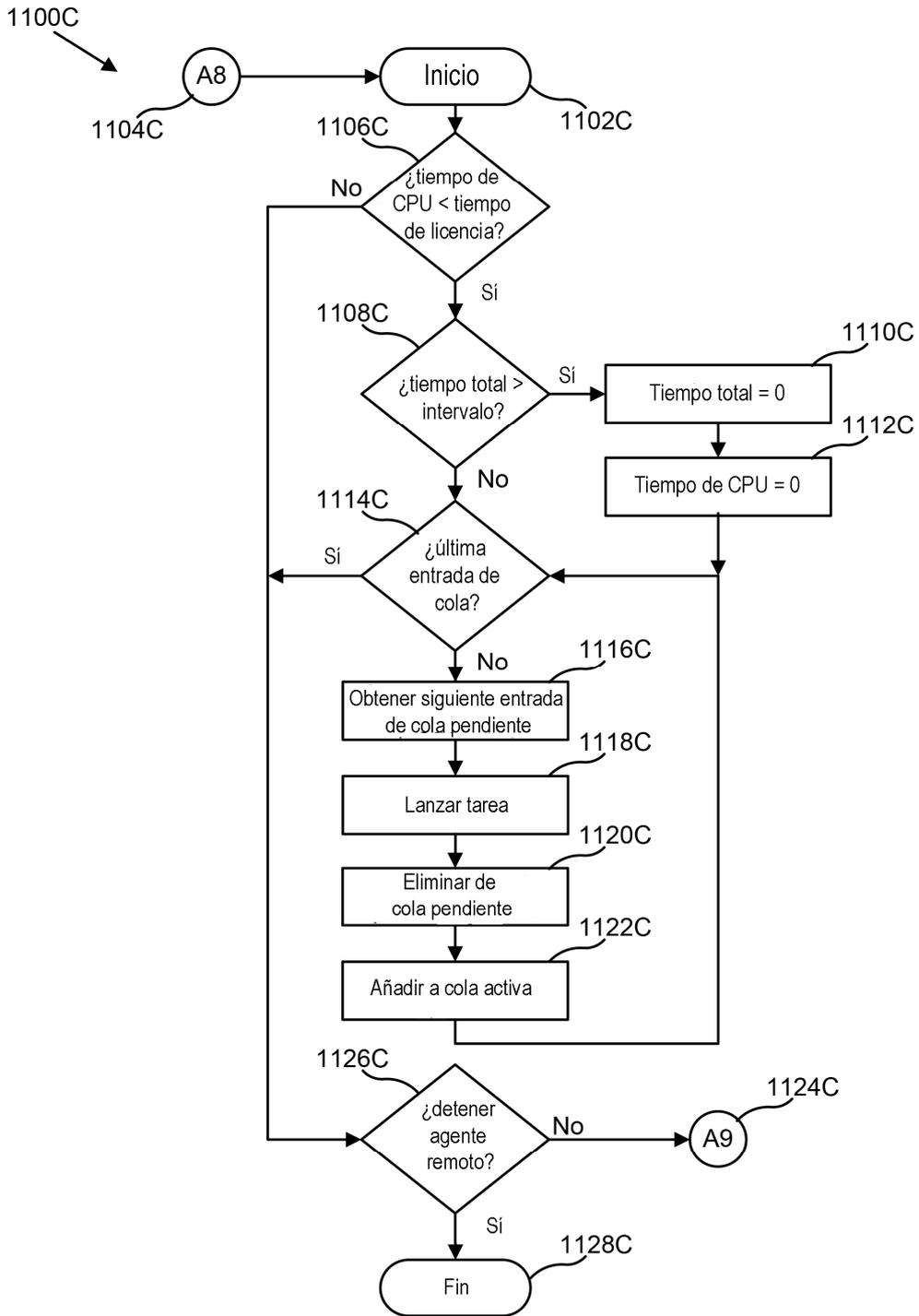


FIG. 11C

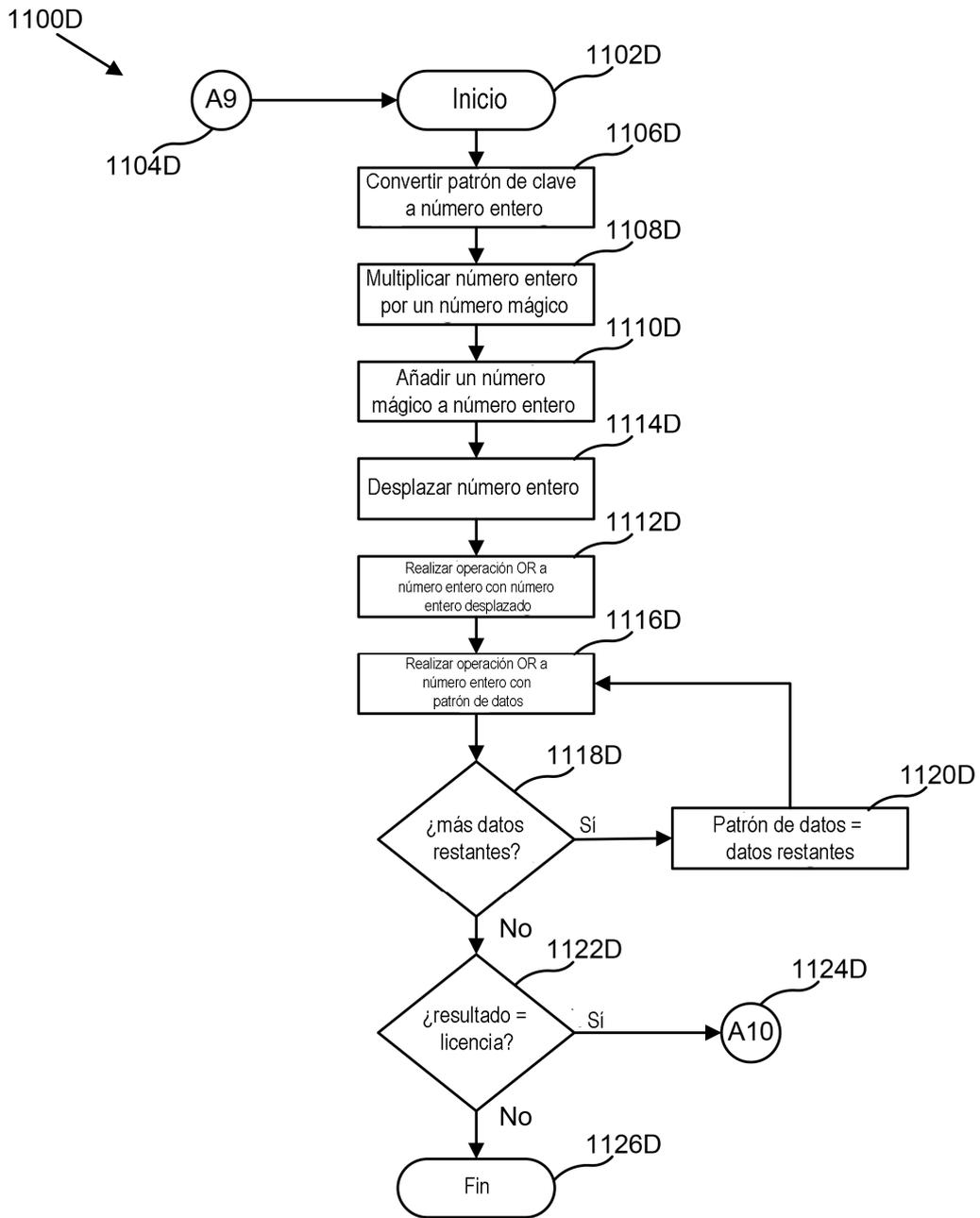


FIG. 11D

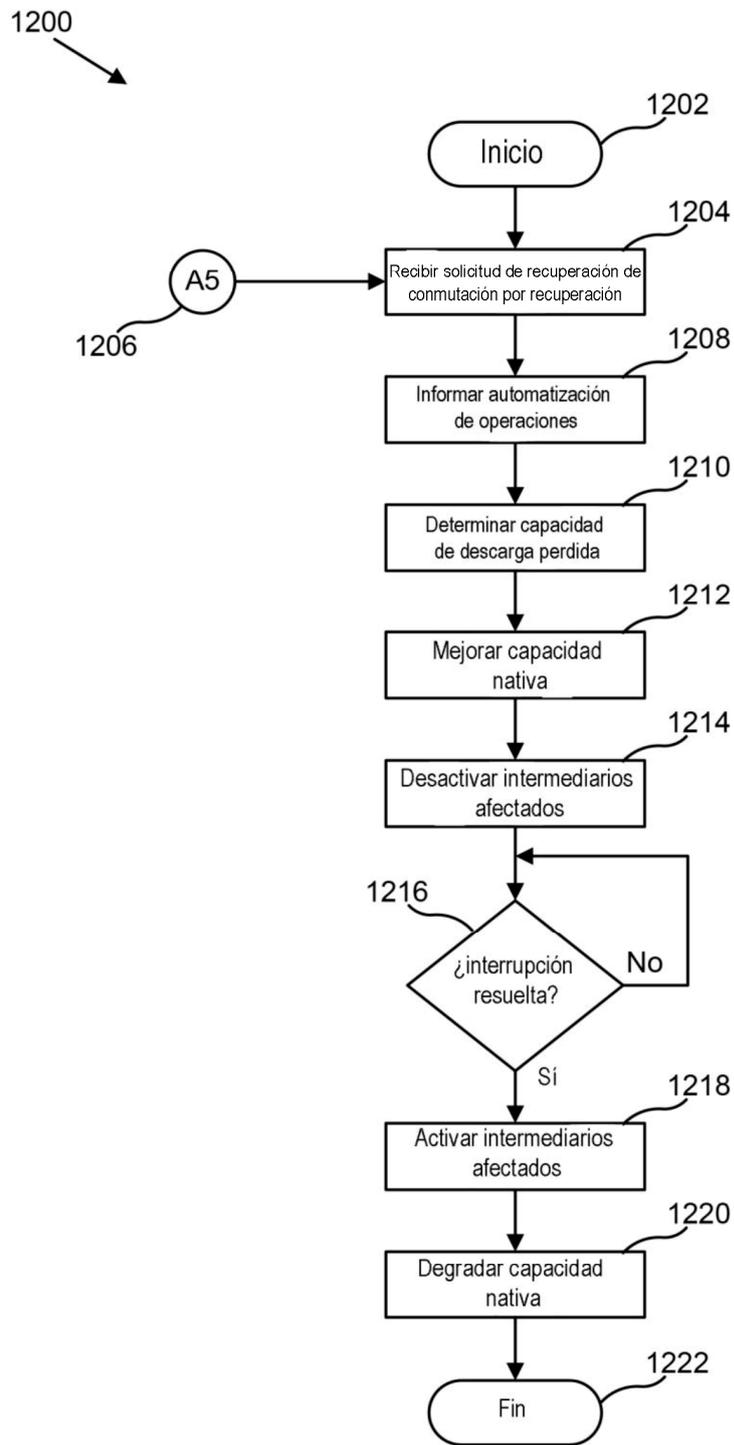


FIG. 12