

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 664 817**

51 Int. Cl.:

G06F 9/455 (2008.01)

G06F 13/10 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **04.11.2004 E 04026219 (8)**

97 Fecha y número de publicación de la concesión europea: **28.02.2018 EP 1548589**

54 Título: **Sistemas y procedimientos de virtualización de dispositivos bimodales de dispositivos basados en hardware reales e idealizados**

30 Prioridad:

12.12.2003 US 734450

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

23.04.2018

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)
One Microsoft Way
Redmond, WA 98052, US**

72 Inventor/es:

TRAUT, ERIC

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 664 817 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Sistemas y procedimientos de virtualización de dispositivos bimodales de dispositivos basados en hardware reales e idealizados

5 La presente invención se refiere en general al campo de los dispositivos en sistemas informáticos y, más en concreto, a dispositivos virtuales potenciados en sistemas informáticos que operan de forma selectiva como un dispositivo virtual de hardware o un dispositivo virtual idealizado, según sea apropiado.

10 Un dispositivo virtual es un dispositivo lógico, que se implementa en software, que se corresponde con algún tipo de dispositivo físico real o idealizado. Existen, por lo general, dos enfoques para modelar dispositivos virtuales: el enfoque de "dispositivos virtuales de hardware", que modela directamente un parte existente de hardware; y el enfoque de "dispositivos virtuales idealizados", que no es un mero reflejo de hardware físico sino que está optimizado para el entorno de VM.

15 El enfoque de dispositivos virtuales de hardware ofrece ventajas con respecto a la compatibilidad - debido a que el dispositivo virtual actúa exactamente como un dispositivo real en todos los aspectos, un software que se ha diseñado para interactuar con ese dispositivo (por ejemplo, un controlador) trabajará con un dispositivo virtual de hardware sin modificación. No obstante, los dispositivos virtuales de hardware se encuentran en desventaja en cuanto al rendimiento - el hardware físico es, a menudo, difícil de emular con un dispositivo virtual sin incurrir en costes generales (e ineficiencias) significativos debido a que, en general, los diseñadores de hardware no tienen en cuenta los problemas de virtualización y, por lo tanto, los dispositivos virtuales de hardware son, a menudo, perceptiblemente más lentos que sus homólogos de hardware reales.

20 Los dispositivos virtuales idealizados, por otro lado, proporcionan una libertad significativa para que los programadores diseñen un dispositivo virtual que sea tanto fácil de implementar como eficiente durante el uso. Debido a que no es necesario que el diseño de un dispositivo virtual idealizado cumpla con las limitaciones impuestas por el diseño de hardware físico, los dispositivos virtuales idealizados se pueden optimizar para uso dentro de un entorno de VM. Además, no es necesario que los programadores de dispositivos virtuales idealizados se preocupen de los efectos secundarios sutiles (tales como temporización, cambios de estado, etc.) en los que se podría basar el software existente para un funcionamiento correcto. Además, los programadores también pueden crear dispositivos virtuales idealizados que son análogos a hardware que, de hecho, no existe, por ejemplo, un dispositivo virtual que permite la comunicación entre un sistema de invitado y un sistema de anfitrión. No obstante, la desventaja es que se pueden plantear problemas de compatibilidad con el enfoque de dispositivos virtuales idealizados debido a que el dispositivo virtual, de hecho, puede no operar exactamente como el dispositivo real en todos los aspectos, y el software que se ha diseñado para interactuar con ese dispositivo físico (por ejemplo, un controlador) puede no funcionar de forma correcta, o no funcionar en absoluto, con un dispositivo virtual idealizado sin modificación.

35 Lo que se necesita en la técnica es un enfoque para los dispositivos virtuales que comprenda las ventajas de los dos enfoques existentes que se analizan en el presente documento pero que carezca de la mayor parte de las limitaciones de los mismos.

40 El documento US 5.548.783 A divulga una controladora de unidad compuesta que incluye un controlador de disco compuesto para soportar un acceso de unidad compuesta. La controladora de disposición de unidades divulgada sirve como una interfaz tanto entre dispositivos de SCSI autónomos como entre dispositivos de SCSI que forman una unidad compuesta. La controladora de disposición de unidades está acoplada entre un ordenador de anfitrión y una pluralidad de dispositivos periféricos de SCSI. La controladora de disposición de unidades incluye una interfaz de emulación de AHA que está acoplada con un controlador de dispositivo de AHA y una interfaz nativa que está acoplada con un controlador de dispositivo compuesto. Por lo tanto, la controladora de disposición de unidades proporciona una interfaz entre un bus de ampliación de anfitrión y una pluralidad de dispositivos periféricos de SCSI.

45 Dos o más de los dispositivos periféricos de SCSI son unidades de disco duro de SCSI que forman, de manera colectiva, una unidad compuesta. Los dispositivos periféricos de SCSI restantes son dispositivos de SCSI autónomos. Los dispositivos periféricos de SCSI se pueden accionar a través de un primer controlador de dispositivo. Como alternativa, las unidades de disco duro de SCSI que forman una unidad compuesta se pueden accionar a través de un segundo controlador de dispositivo que proporciona comandos de acceso de unidad compuesta no de SCSI. El segundo controlador de dispositivo está adaptado para permitir que varias unidades de disco duro dentro de la unidad compuesta atiendan las solicitudes de forma concurrente. La controladora de disposición de unidades incluye dos modos de funcionamiento. El primer modo es un modo de alto rendimiento diseñado para su uso en plataformas que incluyen controladores de dispositivo que se escriben para accionar una unidad compuesta. Se hace referencia al segundo modo como modo de emulación de AHA, que da lugar a que la

55 controladora de disposición de unidades emule una controladora AHA-1540 de Adaptec.

El documento EP-A2-0 664 507 divulga un procedimiento y sistema para proporcionar controladores de dispositivo de modo protegido que son compatibles con controladores de dispositivo de modo real que acceden al mismo dispositivo. Durante el funcionamiento del sistema operativo en modo protegido, los controladores de dispositivo de modo protegido acceden a los dispositivos cuando así sea solicitado por el sistema operativo o un programa de

aplicación. También se proporcionan controladores de dispositivo de modo real que acceden a los dispositivos en modo real cuando sea solicitado por un sistema operativo o programa de aplicación que requiera un funcionamiento de modo real. Los controladores de dispositivo incluyen controladores de adaptador que controlan los adaptadores a través de los cuales se accede a los dispositivos periféricos. El sistema operativo incluye controladores de dispositivo de modo real para los dispositivos periféricos. El sistema operativo también incluye controladores de dispositivo de modo protegido, que controlan los mismos dispositivos que los controladores de dispositivo de modo real.

Por lo tanto, el objeto de la presente invención es la provisión de un sistema informático que comprende un sistema operativo de invitado y un sistema operativo de anfitrión y que mejora el rendimiento de dispositivos virtuales dentro del sistema informático, así como un procedimiento y un medio legible por ordenador correspondientes.

Este objeto se soluciona por medio de la materia objeto de las reivindicaciones independientes.

Algunas formas de realización preferidas se definen por medio de las reivindicaciones dependientes.

Diversas formas de realización de la presente invención están dirigidas a enfoques de dispositivos virtuales bimodales (es decir, "dispositivos bimodales") que combinan las fortalezas relativas del hardware y de los enfoques de virtualización de dispositivos idealizados al tiempo que también mitigan sus debilidades respectivas. En determinadas formas de realización, el dispositivo bimodal es un dispositivo virtual que se basa, principalmente, en una parte real de hardware para proporcionar un amplio grado de compatibilidad con software que se está ejecutando en el entorno de invitado (de forma similar al enfoque de virtualización de dispositivos de hardware). No obstante, para superar el problema de un rendimiento pobre que azota a los dispositivos virtuales de hardware, estas formas de realización también proporcionan un "modo de alto rendimiento" idealizado que no se encuentra en el dispositivo basado en hardware original. Los controladores de software (y otro software) desarrollados para interactuar con el dispositivo de hardware original y que desconoce (y es incapaz de usar) el modo de alto rendimiento continuarán usando el "modo heredado" (virtualización de hardware), mientras que las versiones potenciadas de software de invitado serán capaces de reconocer y usar el modo de alto rendimiento (virtualización idealizada).

Breve descripción de los dibujos

El resumen anterior, así como la siguiente descripción detallada de las formas de realización preferidas, se entienden mejor cuando se leen junto con los dibujos adjuntos. Por razones de ilustración de la invención, se muestran en los dibujos algunas construcciones a modo de ejemplo de la invención; no obstante, la invención no se limita a los medios y procedimientos específicos que se divulgan. En los dibujos:

- la figura 1 es un diagrama de bloques que representa un sistema informático en el que se pueden incorporar algunos aspectos de la invención;
- la figura 2 ilustra la estratificación lógica de la arquitectura de hardware y de software para un entorno operativo emulado en un sistema informático;
- la figura 3A ilustra un sistema informático virtualizado;
- la figura 3B ilustra una forma de realización alternativa de un sistema informático virtualizado que comprende un monitor de máquina virtual que se está ejecutando junto con un sistema operativo de anfitrión;
- la figura 4A ilustra el sistema informático virtualizado de la figura 3 que comprende adicionalmente controladores, dispositivos virtuales y hardware;
- la figura 4B ilustra la trayectoria operativa desde una aplicación que se está ejecutando en un sistema operativo de invitado hasta un dispositivo de hardware en el hardware informático de la figura 4A;
- la figura 5 ilustra de manera más simple la misma interrelación entre los controladores, el dispositivo virtual y el hardware tal como se muestra en la figura 4B;
- la figura 6A ilustra la sustitución de un dispositivo bimodal para el dispositivo virtual en el subsistema de la figura 5, operando dicho dispositivo bimodal en un modo heredado; y
- la figura 6B ilustra el dispositivo bimodal de la figura 6A operando en un modo de alto rendimiento.

Descripción detallada

La materia objeto de la invención se describe de forma específica para cumplir con requisitos estatuarios. No obstante, la descripción no tiene por objeto limitar, por sí misma, el ámbito de la presente patente. Más bien, el inventor de la presente invención ha contemplado que la materia objeto que se reivindica también se podría materializar de otras formas, para incluir diferentes etapas o combinaciones de etapas similares a otras que se describen en el presente documento, junto con otras tecnologías presentes o futuras. Además, a pesar de que el término "etapa" se puede usar en el presente documento para connotar diferentes elementos de procedimientos empleados, no se ha de interpretar que el término implique orden particular alguno entre los mismos o entre diversas etapas que se describen en el presente documento a menos que, y excepto cuando, el orden de las etapas individuales se describa de forma explícita.

Entorno informático

Numerosas formas de realización de la presente invención se pueden ejecutar en un ordenador. La figura 1 y el siguiente análisis tienen por objeto proporcionar una breve descripción general de un entorno informático adecuado en el que se puede implementar la invención. A pesar de que no se requiere, la invención se describirá en el contexto general de las instrucciones ejecutables por ordenador, tales como módulos de programa, que se ejecutan por medio de un ordenador, tal como una estación de trabajo de cliente o un servidor. En general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, y similares para realizar tareas o implementar tipos de datos abstractos. Además, los expertos en la materia apreciarán que la invención se puede poner en práctica con otras configuraciones de sistema, incluyendo dispositivos de mano, sistemas de múltiples procesadores, electrónica de consumo programable o basada en microprocesadores, PC en red, miniordenadores, ordenadores de gran sistema, y similares. La invención también se puede poner en práctica en entornos informáticos distribuidos en los que las tareas se llevan a cabo por medio de dispositivos de procesamiento remoto que están vinculados a través de una red de comunicaciones. En un entorno informático distribuido, los módulos de programa pueden estar ubicados en dispositivos de almacenamiento tanto locales como remotos.

Tal como se muestra en la figura 1, un sistema informático de propósito general a modo de ejemplo incluye un ordenador personal 20, o similar, que incluye una unidad de procesamiento 21, una memoria de sistema 22 y un bus de sistema 23 que acopla diversos componentes de sistema, incluyendo la memoria de sistema a la unidad de procesamiento 21. El bus de sistema 23 puede ser cualquiera de diversos tipos de estructuras de bus o controlador de memoria, bus periférico, y un bus local, usando cualquiera de una diversidad de arquitecturas de buses. La memoria de sistema incluye la memoria de solo lectura (ROM, *read only memory*) 24 y la memoria de acceso aleatorio (RAM, *random access memory*) 25. Un sistema básico de entrada / salida 25 (BIOS, *basic input / output system*), que contiene rutinas básicas que ayudan a transferir la información entre elementos dentro del ordenador personal 20, tal como durante el arranque, se almacenan en la ROM 24. El ordenador personal 20 puede incluir adicionalmente una unidad de disco duro 27 para leer a partir de o escribir en un medio disco duro, que no se muestra, una unidad de disco magnético 28 para leer a partir de o escribir en un disco magnético extraíble 29, y una unidad de disco óptico 30 para leer a partir de o escribir en un disco óptico extraíble 31, tal como un CD ROM, u otro medio óptico. La unidad de disco duro 27, la unidad de disco magnético 28, y la unidad de disco óptico 30 están conectadas al bus de sistema 23 a través de una interfaz de unidad de disco duro 32, una interfaz de disco magnético 33, y una interfaz de disco óptico 34, de forma respectiva. Las unidades y sus medios legibles por ordenador asociados proporcionan un almacenamiento no volátil de instrucciones legibles por ordenador, estructuras de datos, módulos de programa y otros datos para el ordenador personal 20. A pesar de que en el entorno a modo de ejemplo que se describe en el presente documento se emplea un disco duro, un disco magnético extraíble 29 y un disco óptico extraíble 31, ha de ser apreciado por los expertos en la materia que, en el entorno informático a modo de ejemplo, también se pueden usar otros tipos de medios legibles por ordenador que pueden almacenar datos a los que se puede acceder por medio de un ordenador, tales como casetes magnéticos, tarjetas de memoria no volátil, discos de vídeo digital, cartuchos de Bernoulli, memorias de acceso aleatorio (RAM), memorias de solo lectura (ROM), y similares.

Un número de módulos de programa se pueden almacenar en el disco duro, el disco magnético 29, el disco óptico 31, la ROM 24 o la RAM 25, incluyendo un sistema operativo 35, uno o más programas de aplicación 36, otros módulos de programa 37 y datos de programa 38. Un usuario puede capturar comandos e información en el ordenador personal 20 por medio de dispositivos de entrada tales como un teclado 40 y un dispositivo de apuntamiento 42. Otros dispositivos de entrada (que no se muestran) pueden incluir un micrófono, una palanca de mando, un controlador para juegos, una antena parabólica, y similares. Por lo general, estos y otros dispositivos de entrada están conectados a la unidad de procesamiento 21 a través de una interfaz de puerto serie 46 que está acoplada al bus de sistema, pero pueden estar conectados a través de otras interfaces, tales como un puerto paralelo, un puerto para juegos o un bus serie universal (USB, *universal serial bus*). Un monitor 47 u otro tipo de dispositivo de representación también está conectado al bus de sistema 23 a través de una interfaz, tal como un adaptador de vídeo 48. Además del monitor 47, los ordenadores personales incluyen, por lo general, otros dispositivos periféricos de salida (que no se muestran) tales como altavoces e impresoras. El sistema a modo de ejemplo de la figura 1 también incluye un adaptador de anfitrión 55, un bus de pequeña interfaz de sistema informático (SCSI, *Small Computer System Interface*) 56 y un dispositivo de almacenamiento externo 62 que está conectado al bus de SCSI 56.

El ordenador 20 puede operar en un entorno conectado en red usando conexiones lógicas a uno o más ordenadores remotos, tal como el ordenador remoto 49. El ordenador remoto 49 puede ser otro ordenador personal, un servidor, un encaminador, un ordenador en red, un dispositivo del mismo nivel u otro nodo de red común y, por lo general, incluye muchos o todos los elementos que se han descrito en lo que antecede en relación con el ordenador personal 20, a pesar de que solo se ha ilustrado un dispositivo de memoria 50 en la figura 1. Las conexiones lógicas que se describen en la figura 1 incluyen una red de área local (LAN, *local area network*) 51, y una red de área extensa (WAN, *wide area network*) 52. Dichos entornos conectados en red son habituales en oficinas, redes informáticas extensas de empresa, intranets e Internet.

Cuando se implementa en un entorno conectado en red de LAN, el ordenador personal 20 está conectado a la LAN 51 a través de una interfaz o adaptador de red 53. Cuando se implementa en un entorno conectado en red de WAN,

el ordenador personal 20 incluye, por lo general, un módem 54 u otros medios para establecer comunicaciones a través de la red de área extensa 52, tal como Internet. El módem 54, que puede ser interno o externo, puede estar conectado al bus de sistema 23 a través de una interfaz de puerto serie 46. En un entorno conectado en red, los módulos de programa que se describen en relación con el ordenador personal 20, o porciones de los mismos, se pueden almacenar en el dispositivo de almacenamiento de memoria remoto. Se apreciará que las conexiones en red que se muestran son a modo de ejemplo y que se pueden usar otros medios para establecer enlaces de comunicaciones entre los ordenadores. Además, a pesar de que se prevé que numerosas formas de realización de la presente invención son particularmente convenientes para sistemas computarizados, nada en el presente documento tiene por objeto limitar la invención a tales formas de realización.

10 **Máquinas virtuales**

Desde una perspectiva conceptual, los sistemas informáticos, por lo general, comprenden una o más capas de software que se están ejecutando en una capa de hardware fundamental. Esta estratificación se hace por razones de abstracción. Al definir la interfaz para una capa dada de software, la capa puede ser implementada de manera diferente por otras capas por encima de la misma. En un sistema informático bien diseñado, cada capa solo tiene conocimiento acerca de (y solo se basa en) la capa inmediata por debajo de la misma. Esto permite que una capa o una "pila" (múltiples capas colindantes) se reemplace sin afectar de forma negativa a las capas por encima de dicha capa o pila. Por ejemplo, las aplicaciones de software (capas superiores) se basan, por lo general, en niveles más bajos del sistema operativo (capas inferiores) para escribir archivos en alguna forma de almacenamiento permanente, y no es necesario que estas aplicaciones entiendan las diferencias entre los datos escritos en un disco flexible, un disco duro o una carpeta de red. Si esta capa inferior se reemplaza con componentes del nuevo sistema operativo para escribir archivos, la operación de las aplicaciones de software de la capa superior permanece sin alterar.

La flexibilidad del software estratificado permite que una máquina virtual (VM, *virtual machine*) presente una capa de hardware virtual que es, de hecho, otra capa de software. De esta forma, una VM puede crear la ilusión para las capas de software por encima de la misma, de que dichas capas de software se están ejecutando en su propio sistema informático privado y, por lo tanto, las VM pueden permitir que múltiples "sistemas de invitado" se ejecuten de forma concurrente en un "sistema de anfitrión" individual.

La figura 2 es un diagrama que representa la estratificación lógica de la arquitectura de hardware y de software para un entorno operativo emulado en un sistema informático. Un programa de emulación 94 se ejecuta en un sistema operativo de anfitrión y / o arquitectura de hardware 92. El programa de emulación 94 emula una arquitectura de hardware de invitado 96 y un sistema operativo de invitado 98. La aplicación de software 100 se ejecuta, a su vez, en un sistema operativo de invitado 98. En el entorno operativo emulado de la figura 2, debido a la operación del programa de emulación 94, la aplicación de software 100 se puede ejecutar en el sistema informático 90 incluso si la aplicación de software 100 está diseñada para ejecutarse en un sistema operativo que, en general, es incompatible con el sistema operativo de anfitrión y la arquitectura de hardware 92.

La figura 3A ilustra un sistema informático virtualizado que comprende una capa de software de sistema operativo de anfitrión 104 que se está ejecutando directamente por encima del hardware informático físico 102, y el sistema operativo de anfitrión (SO de anfitrión) 104 virtualiza todos los recursos de la máquina al exponer interfaces que son las mismas que el hardware que está virtualizando el SO de anfitrión (lo que posibilita que el SO de anfitrión pase desapercibido para las capas de sistema operativo que se están ejecutando por encima del mismo).

Como alternativa, una capa de software de monitor de máquina virtual, o VMM (*virtual machine monitor*), 104' se puede estar ejecutando en lugar de o junto con un sistema operativo de anfitrión 104", ilustrándose esta última opción en la figura 3B. Por razones de simplicidad, todos los análisis en lo sucesivo en el presente documento (en concreto con respecto al sistema operativo de anfitrión 104) se dirigirán a la forma de realización ilustrada en la figura 3A; no obstante, cada aspecto de dicho análisis será igualmente de aplicación a la forma de realización de la figura 3B, en la que el VMM 104' de la figura 3B reemplaza en esencial, a un nivel funcional, el papel del sistema operativo de anfitrión 104 de la figura 3A que se describe en lo sucesivo en el presente documento.

Haciendo referencia de nuevo a la figura 3A, el SO de anfitrión 104 anterior (o el VMM 104') son dos implementaciones de máquina virtual (VM), VM A 108, que puede ser, por ejemplo, un procesador Intel 386 virtualizado, y VM B 110, que puede ser, por ejemplo, una versión virtualizada de uno de los procesadores de la familia 680X0 de Motorola. Por encima de cada VM 108 y 110 se encuentran los sistemas operativos de invitado (SO de invitado) A 112 y B 114, de forma respectiva. Por encima del SO de invitado A 112 se están ejecutando dos aplicaciones, la aplicación A1 116 y la aplicación A2 118, y por encima del SO de invitado B 114 se encuentra la aplicación B1 112.

55 **Dispositivos virtuales**

Con respecto a los dispositivos de hardware físicos, las VM presentan diversas opciones para permitir que las aplicaciones de software usen dichos dispositivos de hardware. En algunos sistemas de VM, un dispositivo de hardware (tal como un disco duro o un adaptador de red) se puede asignar a una VM individual, y solo el software

que se está ejecutando en esa VM puede usar ese dispositivo de hardware (una arquitectura de dispositivo dedicada). No obstante, este diseño evita la compartición de dispositivos de hardware entre diferentes VM así como también entre una VM y el sistema de anfitrión. Por esta razón, muchas implementaciones de VM usan “dispositivos virtuales” en su lugar.

5 Un dispositivo virtual es un dispositivo lógico, que se implementa en software, que se corresponde con algún tipo de dispositivo físico real o idealizado. En general, los dispositivos virtuales poseen su propio conjunto de estados de dispositivo que puede incluir, por ejemplo, configuraciones de registro, datos almacenados en la memoria intermedia, colas de comandos pendientes, etc. A modo de ejemplo no limitado, para un adaptador de red virtual, el software que está asociado con el adaptador virtual se invocaría cuando el código que se está ejecutando accediera al adaptador de red en la VM, y el adaptador virtual podría entonces responder a comandos de una forma que fuera consistente con una tarjeta de adaptador real. De esta forma, el código que opera dentro de la VM ignora que está “hablando” con un adaptador virtual en vez de con uno real, y el adaptador virtual podría entonces interaccionar directa o indirectamente con el adaptador real para dar lugar a que el adaptador real opere según se desee.

15 En muchos casos, las solicitudes a un dispositivo virtual son asignadas en un dispositivo de anfitrión correspondiente. Por ejemplo, los paquetes de red que se envían a partir de un adaptador de red virtual se pueden encaminar a uno de los adaptadores de red reales instalados en el anfitrión. A este respecto, una arquitectura de dispositivo virtual proporciona una capacidad similar a la arquitectura de dispositivo dedicado que se ha descrito en lo que antecede. No obstante, a diferencia de la arquitectura de dispositivo dedicado, una arquitectura de dispositivo virtual permite que múltiples dispositivos virtuales se asignen en un dispositivo de anfitrión individual (dispositivo real). Por ejemplo, tres VM independientes podrían contener, cada una, un adaptador de red virtual, y los tres adaptadores virtuales podrían, todos ellos, compartir un adaptador de anfitrión individual, mientras que en una arquitectura de dispositivo dedicado solo una VM podría usar el adaptador de anfitrión.

25 La figura 4A ilustra el sistema informático virtualizado de la figura 3, que comprende adicionalmente controladores, dispositivos virtuales y hardware. El SO de invitado A 112 comprende el controlador A 122 y la máquina virtual A 108 comprende el dispositivo virtual A 124 que se corresponde con el controlador A 122. De la misma forma, el SO de invitado B 114 comprende el controlador B 126 y la máquina virtual B 110 comprende el dispositivo virtual B 128 que se corresponde con el controlador B 126. Además, el SO de anfitrión 104 comprende el controlador X 130, y el hardware informático 102 comprende el dispositivo de hardware X 132 que se corresponde con el controlador X 130.

30 La figura 4B ilustra la trayectoria operativa desde una aplicación que se está ejecutando en un sistema operativo de invitado hasta un dispositivo de hardware en el hardware informático. La aplicación A1 116 interacciona 134 con el controlador A 122 que, a su vez, interacciona 136 con el dispositivo virtual A 124. Este dispositivo virtual A 124 interacciona entonces 138 con el controlador X 130 (en el SO de anfitrión 104) que, a su vez, se comunica 140 directamente con el dispositivo de hardware X 132. La figura 5 ilustra de forma más simple la misma interrelación entre los controladores, el dispositivo virtual y el hardware tal como se muestra en la figura 4B.

35 **Dispositivo bimodal**

Por lo general, hay dos enfoques para modelar dispositivos virtuales: el enfoque de “dispositivos virtuales de hardware” que modela directamente una parte existente de hardware; y el enfoque de “dispositivos virtuales idealizados” que no es un mero reflejo del hardware físico sino que está optimizado para el entorno de VM. El enfoque de dispositivos virtuales de hardware ofrece ventajas con respecto a la compatibilidad, debido a que el dispositivo virtual actúa exactamente como un dispositivo real en todos los aspectos, el software que se ha diseñado para interaccionar con ese dispositivo (por ejemplo, un controlador) trabajará con un dispositivo de hardware virtual sin modificación. No obstante, los dispositivos virtuales de hardware se encuentran en desventaja en cuanto al rendimiento - el hardware físico es, a menudo, difícil de emular con un dispositivo virtual sin incurrir en costes generales (e ineficiencias) significativos debido a que, en general, los diseñadores de hardware no tienen en cuenta los problemas de virtualización y, por lo tanto, los dispositivos virtuales de hardware son, a menudo, perceptiblemente más lentos que sus homólogos de hardware reales.

Los dispositivos virtuales idealizados, por otro lado, proporcionan una libertad significativa para que los programadores diseñen un dispositivo virtual que sea tanto fácil de implementar como eficiente durante el uso. Debido a que no es necesario que el diseño de dispositivos virtuales idealizados cumpla con las limitaciones impuestas por el diseño de hardware físico, los dispositivos virtuales idealizados se pueden optimizar para uso dentro de un entorno de VM. Además, no es necesario que los programadores de dispositivos virtuales idealizados se preocupen de los efectos secundarios sutiles (tales como temporización, cambios de estado, etc.) en los que se podría basar el software existente para un funcionamiento correcto. Además, los programadores también pueden crear dispositivos virtuales idealizados que son análogos a hardware que, de hecho, no existe, por ejemplo, un dispositivo virtual que permite la comunicación entre un sistema de invitado y un sistema de anfitrión. No obstante, la desventaja es que se pueden plantear problemas de compatibilidad con el enfoque de dispositivos virtuales idealizados debido a que el dispositivo virtual, de hecho, puede no operar exactamente como el dispositivo real en todos los aspectos, y el software que se ha diseñado para interaccionar con ese dispositivo físico (por ejemplo, un controlador) puede no funcionar de forma correcta, o no funcionar en absoluto, con un dispositivo virtual idealizado sin modificación.

Diversas formas de realización de la presente invención están dirigidas a enfoques de dispositivos virtuales bimodales (es decir, “dispositivos bimodales”) que combinan las fortalezas relativas del hardware y de los enfoques de virtualización de dispositivos idealizados al tiempo que también mitigan sus debilidades respectivas. En determinadas formas de realización, el dispositivo bimodal es un dispositivo virtual que se basa, principalmente, en una parte real de hardware para proporcionar un amplio grado de compatibilidad con software que se está ejecutando en el entorno de invitado (de forma similar al enfoque de virtualización de dispositivos de hardware). No obstante, para superar el problema de un rendimiento pobre que azota a los dispositivos virtuales de hardware, estas formas de realización también proporcionan un “modo de alto rendimiento” idealizado que no se encuentra en el dispositivo basado en hardware original. Los controladores de software (y otro software) desarrollados para interaccionar con el dispositivo de hardware original y que desconoce (y es incapaz de usar) el modo de alto rendimiento continuarán usando el “modo heredado” (virtualización de hardware), mientras que las versiones potenciadas de software de invitado serán capaces de reconocer y usar el modo de alto rendimiento (virtualización idealizada).

Para determinadas formas de realización, el modo de alto rendimiento puede ser completamente independiente de la funcionalidad original del modo heredado para el dispositivo virtual. Como alternativa, el modo de alto rendimiento puede comprender ampliaciones o reemplazos a la funcionalidad original. En cualquier forma, el modo de alto rendimiento es parecido a los enfoques de virtualización de procesadores idealizados, ya sea independiente, o una mejora, de la funcionalidad del modo heredado. Además, se ha de observar que, dependiendo de qué forma de realización se use, el cambio al modo de alto rendimiento puede deshabilitar completamente el modo heredado, puede deshabilitar porciones del modo heredado y / o puede ampliar la funcionalidad del modo heredado.

En determinadas formas de realización de la presente invención, el modo de alto rendimiento se puede habilitar a través del uso de bits reservados dentro de los registros del dispositivo virtual, mientras que las formas de realización alternativas podrían añadir en su lugar nuevos registros (por ejemplo, un registro creado específicamente para su utilización por uno o más dispositivos virtuales) o puede implementar una forma de saludo inicial a través del que se escribe una secuencia preestablecida de comandos o datos en los registros heredados.

A modo de ejemplo no limitado, considérese un dispositivo bimodal NIC de Ethernet. En el modo heredado, el envío de un paquete de información a la tarjeta de Ethernet podría involucrar hasta ocho (8) escrituras individuales en los registros de E/S heredados. Estos accesos de E/S son requeridos para configurar parámetros para la transferencia de paquetes, incluyendo la dirección DMA (es decir, la ubicación en RAM en la que se pueden encontrar los datos de paquete), la longitud de la transferencia y un byte de comando que inicia la transferencia una vez que se han especificado otros parámetros. No obstante, cada una de las escrituras en estos registros de E/S es relativamente costosa dentro de una VM, y el tiempo combinado que se requiere para virtualizar estos accesos de E/S es significativo y da como resultado una pérdida importante de rendimiento. No obstante, una tarjeta de Ethernet virtual “idealizada”, es decir, una tarjeta virtual que se está ejecutando en el modo de alto rendimiento, podría requerir solo un acceso de E/S para iniciar un envío de paquetes y, por lo tanto, un dispositivo bimodal de tarjeta de Ethernet (por ejemplo, uno basado en un conjunto de chips DEC 21140) que se ampliase con el modo de alto rendimiento podría incrementar de manera significativa el rendimiento en aquellas instancias en las que se pudiera usar el modo de alto rendimiento.

Para determinadas formas de realización, el modo de alto rendimiento se podría habilitar a través del uso de un controlador modificado, especial instalado dentro del entorno de sistema operativo de invitado y, una vez que se ha habilitado el modo de alto rendimiento, el controlador puede usar el mecanismo más óptimo para iniciar un envío de paquetes. No obstante, si el controlador modificado no se encuentra presente en el invitado, un controlador heredado seguiría siendo capaz de proporcionar una funcionalidad conexión de red, aunque sin la optimización de rendimiento que es permitida por el modo de alto rendimiento.

La figura 6A ilustra la sustitución de un dispositivo bimodal para el dispositivo virtual en el subsistema de la figura 5, operando dicho dispositivo bimodal en un modo heredado. En esta figura, un controlador heredado A' 122' interacciona 136' con el dispositivo bimodal 150 que comprende tanto un modo heredado 152 para la virtualización de hardware como un modo de alto rendimiento 154 para la virtualización idealizada. Debido a que el controlador heredado A' 122' no está diseñado para trabajar con la virtualización idealizada que es proporcionada por el modo de alto rendimiento 154 del dispositivo bimodal 150, este trabaja en su lugar con la virtualización de hardware que es proporcionada por el modo heredado 152 del dispositivo bimodal 150 (tal como se muestra) que, a su vez, interacciona 138' con el controlador de SO de anfitrión X 130.

En contraposición, la figura 6B ilustra el dispositivo bimodal operando en un modo de alto rendimiento. En esta figura, un controlador de alto rendimiento A" 122" interacciona 136" con el dispositivo bimodal 150. Debido a que el controlador de alto rendimiento A" 122" es capaz de trabajar con la virtualización idealizada que es proporcionada por el modo de alto rendimiento 154 del dispositivo bimodal 150, lo hace (tal como se muestra) y el dispositivo bimodal 150, a su vez, interacciona 138" con el controlador de SO de anfitrión X 130.

Los enfoques “bimodales” basados en hardware se han usado en diferentes áreas tecnológicas para añadir funcionalidad a un dispositivo de hardware que es, por lo demás, incompatible con la funcionalidad antigua. Por ejemplo, un ratón de PS/2 tradicional envía tres bytes de información al ordenador (coordenada X, coordenada Y y

estado de botón) pero cuando los fabricantes de ratones añadieron una “rueda de desplazamiento” a los ratones, los mismos también necesitaron cambiar a un paquete de cuatro bytes de información que incluye el estado de la rueda de desplazamiento. Con el fin de proporcionar una compatibilidad con versiones anteriores de controladores de ratón que no tienen conocimiento de ruedas de desplazamiento, el nuevo ratón se iniciaba en el “modo heredado” y solo proporcionaba tres bytes de información, pero cuando se cargaba un controlador de nuevo estilo, este controlador habilitaba el “modo de rueda de desplazamiento” al indicar al ratón que comenzara a enviar paquetes de cuatro bytes.

Técnicas de dispositivo bimodal

Existen diversas técnicas a través de las cuales los controladores de modo de invitado pueden identificar un dispositivo bimodal (es decir, diferenciar entre un dispositivo de hardware real y un dispositivo emulado que soporta modos de operación adicionales). Estas técnicas incluyen, pero no se limitan a, lo siguiente:

- Número de versión de dispositivo único: Muchos dispositivos contienen identificadores de versión de dispositivo a los que puede acceder el software. En ocasiones, se puede acceder a estos valores directamente a través de un puerto de E / S o registro asignado en memoria. Otras veces, se puede enviar un comando específico al dispositivo solicitando que devuelva la información de la versión. En estos casos, en ocasiones es posible modificar el número de versión que no se corresponde con hardware real existente alguno. La mayoría de los controladores ignoran el número de versión o realizan una prueba en busca de un número de versión mínimo, de tal modo que la selección de un número de versión único y nuevo puede ser posible sin afectar a la compatibilidad con los controladores heredados existentes.
- Registros no usados: Algunos dispositivos definen un intervalo de puertos de E / S o registros asignados en memoria pero solo usan un subconjunto del intervalo. Esto es común debido a que los intervalos están asignados, por lo general, en bloques en cantidades de potencias de dos, más el hecho de que, por lo general, los ingenieros asignan, para una ampliación futura, más de lo que es necesario. Es posible usar un registro “no definido” o “reservado” en el intervalo como un identificador. El hardware real devolverá, por lo general, un valor de 0x00 o 0xFF cuando se lee un registro reservado. El dispositivo bimodal emulado podría devolver un valor distinto y diferente.
- Saludo inicial: Si las dos técnicas anteriores no son factibles, habitualmente es posible diseñar alguna forma de “saludo inicial” de puerta trasera. Esto involucra una secuencia particular de accesos a registro o comandos de dispositivo que resulta improbable que se hallen en los escenarios de uso del mundo real. Por ejemplo, si el dispositivo fuera un controlador de disco, una técnica de saludo inicial podría involucrar el envío de una secuencia de comandos que leyese cero bytes iniciando una lista predeterminada y específica de sectores en un orden específico (por ejemplo, 0, 3 y 7). En un controlador real, esto no produciría efecto alguno (debido a que una lectura de cero bytes se define como una no operación). No obstante, en un dispositivo bimodal, una secuencia de este tipo podría dar lugar a algún otro efecto secundario (por ejemplo, que se establezca un bit de estado específico en el registro de estados).

Naturalmente, las mismas técnicas usadas para identificar un dispositivo emulado capaz de la operación bimodal también se pueden usar para habilitar de forma automática el modo “potenciado” para dicho dispositivo.

Conclusión

Los diversos sistemas, procedimientos y técnicas que se describen en el presente documento se pueden implementar con hardware o software o, cuando sea apropiado, con una combinación de ambos. Por lo tanto, los procedimientos y aparatos de la presente invención, o determinados aspectos o porciones de la misma, pueden adoptar la forma de código de programa (es decir, instrucciones) que se materializa en un medio tangible, tal como discos flexibles, CD-ROM, discos duros, o cualquier otro medio de almacenamiento legible por máquina en el que, cuando el código de programa se carga en y es ejecutado por una máquina, tal como un ordenador, la máquina se convierte en un aparato para poner en práctica la invención. En el caso de la ejecución del código del programa en ordenadores programables, en general el ordenador incluirá un procesador, un medio de almacenamiento legible por el ordenador (incluyendo memoria volátil y no volátil y / o elementos de almacenamiento), por lo menos un dispositivo de entrada, y por lo menos un dispositivo de salida. Uno o más programas se implementan preferiblemente en un lenguaje de programación orientado a objetos, o de procedimiento, de alto nivel, para comunicarse con un sistema informático. No obstante, el programa o programas se pueden implementar en un lenguaje ensamblador o máquina, si así se desea. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado, y combinarse con implementaciones de hardware.

Los procedimientos y aparatos de la presente invención también se pueden materializar en la forma de código de programa que se transmite a través de algún medio de transmisión, tal como a través de hilos eléctricos o cableado, a través de fibra óptica o a través de cualquier otra forma de transmisión en la que, cuando el código de programa se recibe y se carga en y es ejecutado por una máquina, tal como una EPROM, una disposición de puertas, un dispositivo lógico programable (PLD, *programmable logic device*), un ordenador de cliente, un aparato de grabación de vídeo o similar, la máquina se convierte en un aparato para poner en práctica la invención. Cuando se implementa en un procesador de propósito general, el código del programa se combina con el procesador para proporcionar un aparato único que opera para realizar la funcionalidad de indexación de la presente invención.

Debido a que la presente invención se ha descrito en conexión con las formas de realización preferidas de las diversas figuras, se entiende que se pueden usar otras formas de realización similares o se pueden hacer modificaciones y adiciones a la forma de realización que se describe para llevar a cabo la misma función de la presente invención sin apartarse de la misma. Por ejemplo, a pesar de que algunas formas de realización a modo de ejemplo de la invención se describen en el contexto de dispositivos digitales que emulan la funcionalidad de ordenadores personales, un experto en la materia reconocerá que la presente invención no se limita a dichos dispositivos digitales, tal como se describe en la presente solicitud se puede aplicar a cualquier número de dispositivos o entornos informáticos existentes o emergentes, tales como una consola de juegos, un ordenador de mano, etc., ya sea cableado o inalámbrico, y se puede aplicar a cualquier número de tales dispositivos informáticos conectados a través de una red de comunicaciones, y que interaccionan a través de la red. Además, se debería resaltar que se contemplan en el presente documento una diversidad de plataformas informáticas, incluyendo sistemas operativos de dispositivos de mano y otros sistemas de interfaz de hardware / software para aplicaciones específicas, especialmente debido a que continúa proliferando el número de dispositivos inalámbricos conectados en red. Por consiguiente, la presente invención no se debería limitar a forma de realización individual alguna, sino más bien interpretarse, en cuanto a su amplitud y ámbito, de acuerdo con las reivindicaciones adjuntas.

Por último, las formas de realización que se describen en el presente documento se pueden adaptar para uso en otras arquitecturas de procesadores, sistemas basados en ordenadores o virtualizaciones de sistemas y dichas formas de realización son anticipadas de forma expresa para las divulgaciones que se hacen en el presente documento y, por lo tanto, la presente invención no se debería limitar a las formas de realización específicas que se describen en el presente documento sino interpretarse en su lugar de la forma más amplia. De la misma forma, también se anticipa, por medio de las divulgaciones que se hacen en el presente documento, el uso de instrucciones sintéticas para fines que no sean la virtualización de procesadores, y cualquier uso de este tipo de las instrucciones sintéticas en contextos que no sean la virtualización de procesadores se ha de interpretar de la forma más amplia en las divulgaciones que se hacen en el presente documento.

25

REIVINDICACIONES

1. Un sistema informático, comprendiendo dicho sistema informático un sistema operativo de invitado (112, 114), un dispositivo virtual bimodal (150) y un sistema operativo de anfitrión (104), en el que dicho sistema operativo de anfitrión comprende una implementación de máquina virtual que comprende dicho sistema operativo de invitado, en el que dicho sistema operativo de invitado comprende un primer controlador (122', 122'') que interacciona con dicho dispositivo virtual bimodal, en el que dicho dispositivo virtual bimodal se implementa en dicha implementación de máquina virtual, y dicho sistema operativo de anfitrión comprende un segundo controlador (130), y en el que dicho dispositivo virtual bimodal interacciona con dicho segundo controlador y opera de forma selectiva como un dispositivo virtual de hardware en un primer modo heredado (153) y como un dispositivo virtual idealizado en un segundo modo de alto rendimiento (154), en el que dicho dispositivo virtual de hardware en dicho primer modo heredado actúa como un dispositivo real al modelar una parte de hardware físico (132) y dicho dispositivo virtual idealizado en dicho segundo modo de alto rendimiento se optimiza para su uso dentro de un entorno de máquina virtual al no ser un mero reflejo de hardware físico y al no tener que cumplir con un diseño de hardware físico.
2. El sistema informático de la reivindicación 1, en el que:
- 15 el dispositivo virtual bimodal (150) opera de forma selectiva como dicho dispositivo virtual de hardware cuando dicho primer controlador (122') que interacciona con dicho dispositivo virtual bimodal no se ha diseñado para interaccionar con dicho dispositivo virtual bimodal que opera en dicho segundo modo de alto rendimiento (154); y el dispositivo virtual bimodal (150) opera de forma selectiva como dicho dispositivo virtual idealizado cuando dicho primer controlador (122'') que interacciona con dicho dispositivo virtual bimodal se ha diseñado para interaccionar con dicho dispositivo virtual bimodal que opera en dicho segundo modo de alto rendimiento (154).
3. El sistema informático de la reivindicación 2, en el que la funcionalidad del segundo modo de alto rendimiento (154) amplía la funcionalidad del primer modo heredado (152).
4. El sistema informático de la reivindicación 2, en el que la funcionalidad del segundo modo de alto rendimiento (154) es independiente de la funcionalidad del primer modo heredado (152).
- 25 5. El sistema informático de la reivindicación 4, en el que la funcionalidad del segundo modo de alto rendimiento (154) deshabilita la funcionalidad del primer modo heredado (152).
6. El sistema informático de la reivindicación 4, en el que la funcionalidad del segundo modo de alto rendimiento (154) deshabilita porciones de la funcionalidad del primer modo heredado (152).
- 30 7. El sistema informático de la reivindicación 2, en el que el segundo modo de alto rendimiento (154) se habilita a través del uso de al menos un bit en un registro de dispositivos virtuales.
8. El sistema informático de la reivindicación 2, en el que el segundo modo de alto rendimiento (154) se habilita a través del uso de al menos un bit en un registro creado específicamente para su utilización por uno o más dispositivos virtuales.
9. El sistema informático de la reivindicación 2, en el que el segundo modo de alto rendimiento (154) se habilita a través del uso de una secuencia preestablecida de comandos o datos que cambian un valor en al menos un registro.
- 35 10. El sistema informático de la reivindicación 2, en el que el segundo modo de alto rendimiento (154) se habilita a través del uso de un segundo controlador de modo (122'') que está instalado dentro del entorno de sistema operativo de invitado; y si no se encuentra presente el segundo controlador de modo, se habilita en su lugar un primer controlador de modo (122').
- 40 11. El sistema informático de una cualquiera de las reivindicaciones 1 a 10, en el que dicho sistema informático comprende un entorno de máquina virtual y dicho dispositivo virtual bimodal (150) que opera de forma selectiva como dicho dispositivo virtual de hardware en dicho primer modo heredado (152) y como dicho dispositivo virtual idealizado en dicho segundo modo de alto rendimiento (154) con dicho entorno de máquina virtual.
- 45 12. Un medio legible por ordenador que comprende instrucciones legibles por ordenador, comprendiendo dichas instrucciones legibles por ordenador instrucciones para un dispositivo virtual bimodal (150) que está comprendido en un sistema informático para operar de forma selectiva como un dispositivo virtual de hardware en un primer modo heredado (152) y como un dispositivo virtual idealizado en un segundo modo de alto rendimiento (154), en el que dicho sistema informático comprende adicionalmente un sistema operativo de invitado (112, 144) y un sistema operativo de anfitrión (104), en el que dicho sistema operativo de anfitrión comprende una implementación de máquina virtual que comprende dicho sistema operativo de invitado, en el que dicho sistema operativo de invitado comprende un primer controlador (122', 122'') que interacciona con dicho dispositivo virtual bimodal, en el que dicho dispositivo virtual bimodal se implementa en dicha implementación de máquina virtual, dicho sistema operativo de anfitrión comprende un segundo controlador (130) y dicho dispositivo virtual bimodal interacciona con dicho segundo controlador; y en el que
- 50
- 55

dicho dispositivo virtual de hardware en dicho primer modo heredado actúa como un dispositivo real al modelar una parte de hardware físico (132) y dicho dispositivo virtual idealizado en dicho segundo modo de alto rendimiento se optimiza para su uso dentro de un entorno de máquina virtual al no ser un mero reflejo de hardware físico y al no tener que cumplir con un diseño de hardware físico.

- 5 13. El medio legible por ordenador de la reivindicación 12, que comprende adicionalmente instrucciones para:
- el dispositivo virtual bimodal para operar de forma selectiva como dicho dispositivo virtual de hardware cuando dicho primer controlador (122') que interacciona con dicho dispositivo virtual bimodal (150) no se ha diseñado para interaccionar con dicho dispositivo virtual bimodal que opera en dicho segundo modo de alto rendimiento (154); y
- 10 el dispositivo virtual bimodal para operar de forma selectiva como dicho dispositivo virtual idealizado cuando dicho primer controlador (122'') que interacciona con dicho dispositivo virtual bimodal (150) se ha diseñado para interaccionar con dicho dispositivo virtual bimodal que opera en dicho segundo modo de alto rendimiento (154).
14. El medio legible por ordenador de la reivindicación 13, que comprende adicionalmente instrucciones para la funcionalidad del segundo modo de alto rendimiento para ampliar la funcionalidad del primer modo heredado (152).
- 15 15. El medio legible por ordenador de la reivindicación 13, que comprende adicionalmente instrucciones para la funcionalidad del segundo modo de alto rendimiento que son independientes y distintas de las instrucciones para la funcionalidad del primer modo heredado (152).
16. El medio legible por ordenador de la reivindicación 15, que comprende adicionalmente instrucciones para el segundo modo de alto rendimiento para deshabilitar la funcionalidad del primer modo heredado (152).
- 20 17. El medio legible por ordenador de la reivindicación 15, que comprende adicionalmente instrucciones para el segundo modo de alto rendimiento para deshabilitar porciones de la funcionalidad del primer modo heredado (152).
18. El medio legible por ordenador de la reivindicación 13, que comprende adicionalmente instrucciones para habilitar el segundo modo de alto rendimiento (154) a través del uso de al menos un bit en un registro de dispositivos virtuales.
- 25 19. El medio legible por ordenador de la reivindicación 13, que comprende adicionalmente instrucciones para habilitar el segundo modo de alto rendimiento (154) a través del uso de al menos un bit en un registro creado específicamente para su utilización por uno o más dispositivos virtuales.
20. El medio legible por ordenador de la reivindicación 13, que comprende adicionalmente instrucciones para habilitar el segundo modo de alto rendimiento (154) a través del uso de una secuencia preestablecida de comandos o datos que cambian un valor en al menos un registro.
- 30 21. El medio legible por ordenador de la reivindicación 13, que comprende adicionalmente instrucciones para:
- habilitar el segundo modo de alto rendimiento (154) a través del uso de un segundo controlador de modo (122'') que está instalado dentro de un entorno de sistema operativo de invitado; y
- 35 si no se encuentra presente el segundo controlador de modo, habilitar un primer modo heredado (152) a través del uso de un primer controlador de modo (122').
22. Un procedimiento de mejora del rendimiento de dispositivos virtuales en un sistema informático, comprendiendo dicho sistema informático un sistema operativo de invitado (112, 114), un dispositivo virtual bimodal (150) y un sistema operativo de anfitrión (104), en el que dicho sistema operativo de anfitrión comprende una implementación de máquina virtual que comprende dicho sistema operativo de invitado, en el que dicho sistema operativo de invitado comprende un primer controlador (122', 122'') que interacciona con dicho dispositivo virtual bimodal, en el que dicho dispositivo virtual bimodal se implementa en dicha implementación de máquina virtual, dicho sistema operativo de anfitrión comprende un segundo controlador (130) y dicho dispositivo virtual bimodal interacciona con dicho segundo controlador,
- 40 comprendiendo dicho procedimiento utilizar dicho dispositivo virtual bimodal para operar de forma selectiva como un dispositivo virtual de hardware en un primer modo heredado (152) y como un dispositivo virtual idealizado en un segundo modo de alto rendimiento (154), en el que dicho dispositivo virtual de hardware en dicho primer modo heredado actúa como un dispositivo real al modelar una parte de hardware físico (132) y dicho dispositivo virtual idealizado en dicho segundo modo de alto rendimiento se optimiza para su uso dentro de un entorno de máquina virtual al no ser un mero reflejo de hardware físico y al no tener que cumplir con un diseño de hardware físico.
- 45 23. El procedimiento de la reivindicación 22, en el que:
- el dispositivo virtual bimodal opera de forma selectiva como dicho dispositivo virtual de hardware cuando dicho primer controlador (122') que interacciona con dicho dispositivo virtual bimodal (150) no se ha diseñado para interaccionar con dicho dispositivo virtual bimodal que opera en dicho segundo modo de alto rendimiento (154); y
- 50 el dispositivo virtual bimodal opera de forma selectiva como dicho dispositivo virtual idealizado cuando dicho

primer controlador (122") que interacciona con dicho dispositivo virtual bimodal se ha diseñado para interactuar con dicho dispositivo virtual bimodal que opera en dicho segundo modo de alto rendimiento (154).

24. El procedimiento de la reivindicación 23, en el que la funcionalidad del segundo modo de alto rendimiento (154) amplía la funcionalidad del primer modo heredado (152).
- 5 25. El procedimiento de la reivindicación 23, en el que la funcionalidad del segundo modo de alto rendimiento (154) es independiente de la funcionalidad del primer modo heredado (152).
26. El procedimiento de la reivindicación 25, en el que la funcionalidad del segundo modo de alto rendimiento (154) deshabilita la funcionalidad del primer modo heredado (152).
- 10 27. El procedimiento de la reivindicación 25, en el que la funcionalidad del segundo modo de alto rendimiento (154) deshabilita porciones de la funcionalidad del primer modo heredado (152).
28. El procedimiento de la reivindicación 23, en el que el segundo modo de alto rendimiento (154) se habilita a través del uso de al menos un bit en un registro de dispositivos virtuales.
- 15 29. El procedimiento de la reivindicación 23, en el que el segundo modo de alto rendimiento (154) se habilita a través del uso de al menos un bit en un registro creado específicamente para su utilización por uno o más dispositivos virtuales.
30. El procedimiento de la reivindicación 23, en el que el segundo modo de alto rendimiento (154) se habilita a través del uso de una secuencia preestablecida de comandos o datos que cambian un valor en al menos un registro.
- 20 31. El procedimiento de la reivindicación 23, en el que el segundo modo de alto rendimiento (154) se habilita a través del uso de un segundo controlador de modo (122") que está instalado dentro de un entorno de sistema operativo de invitado; y si no se encuentra presente el segundo controlador de modo, se habilita en su lugar un primer controlador de modo (122').

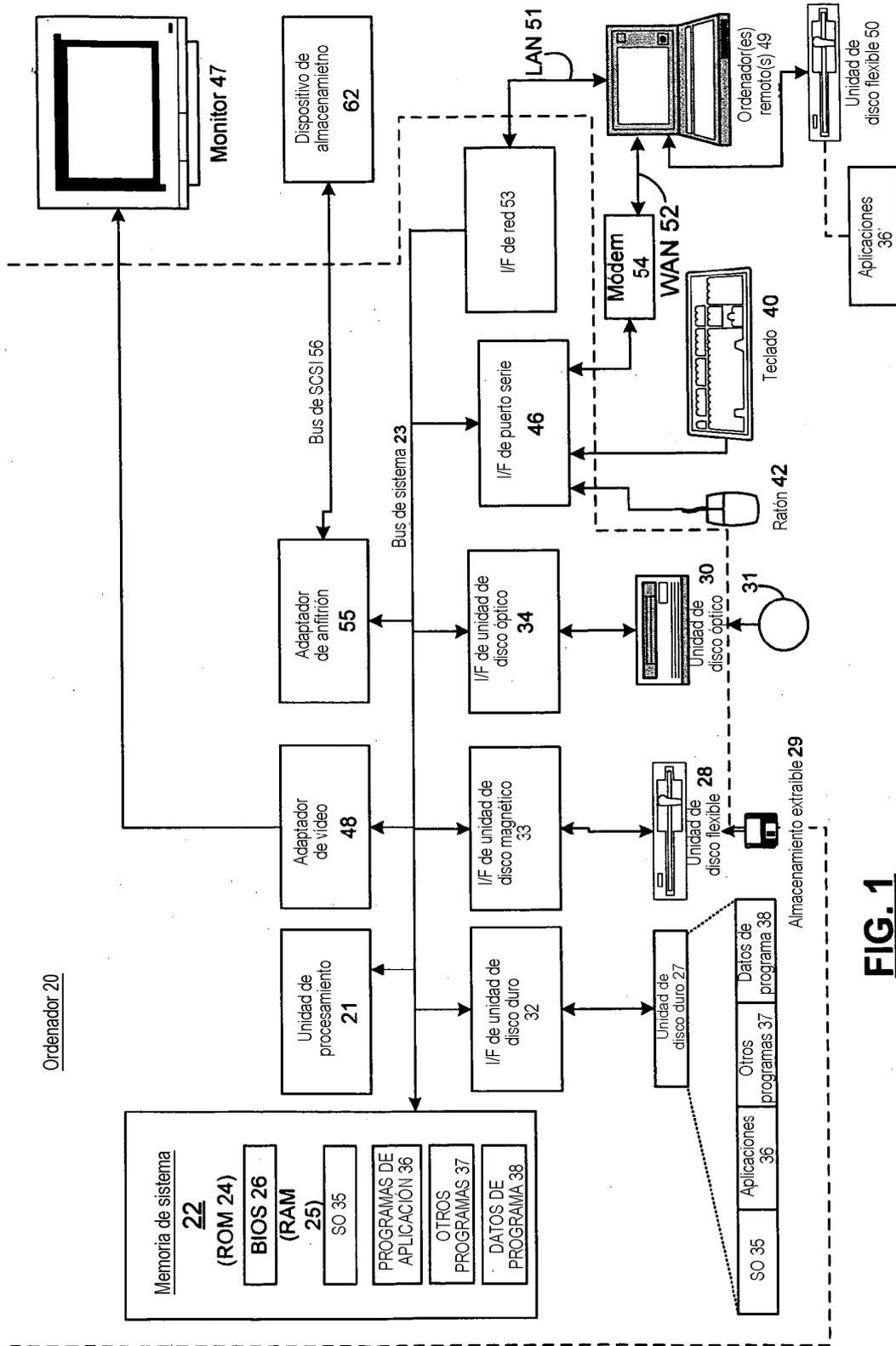


FIG. 1

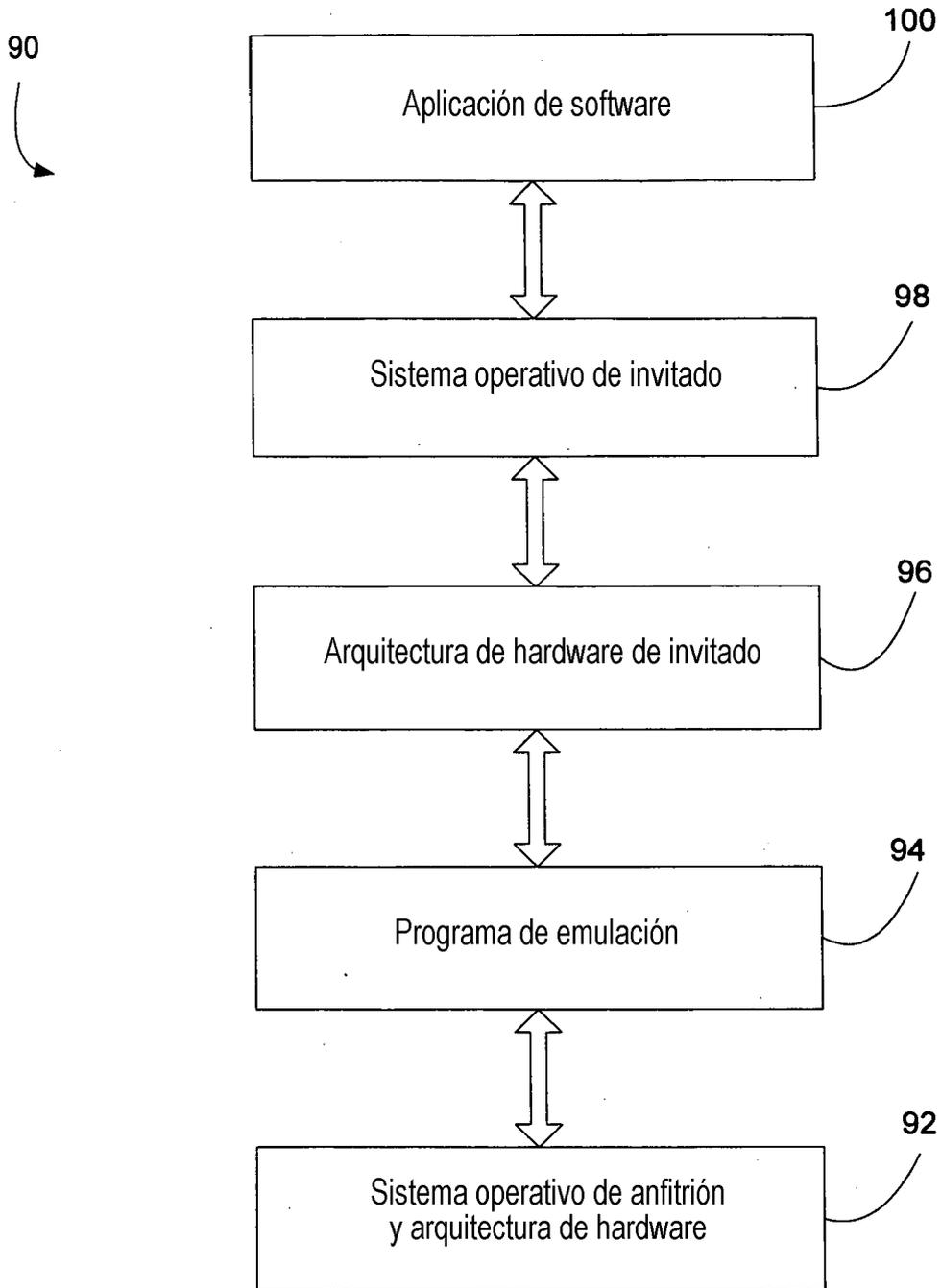


FIG. 2

FIG. 3A

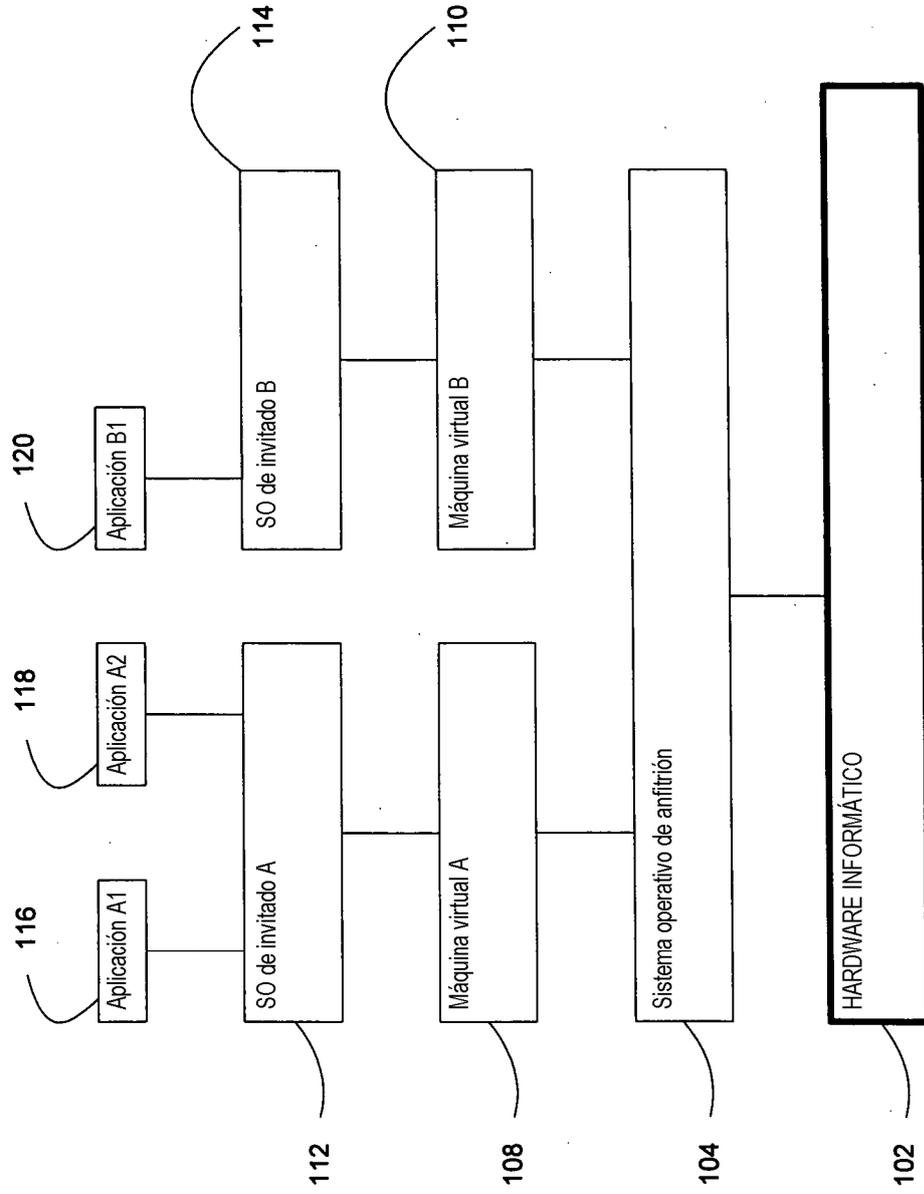


FIG. 3B

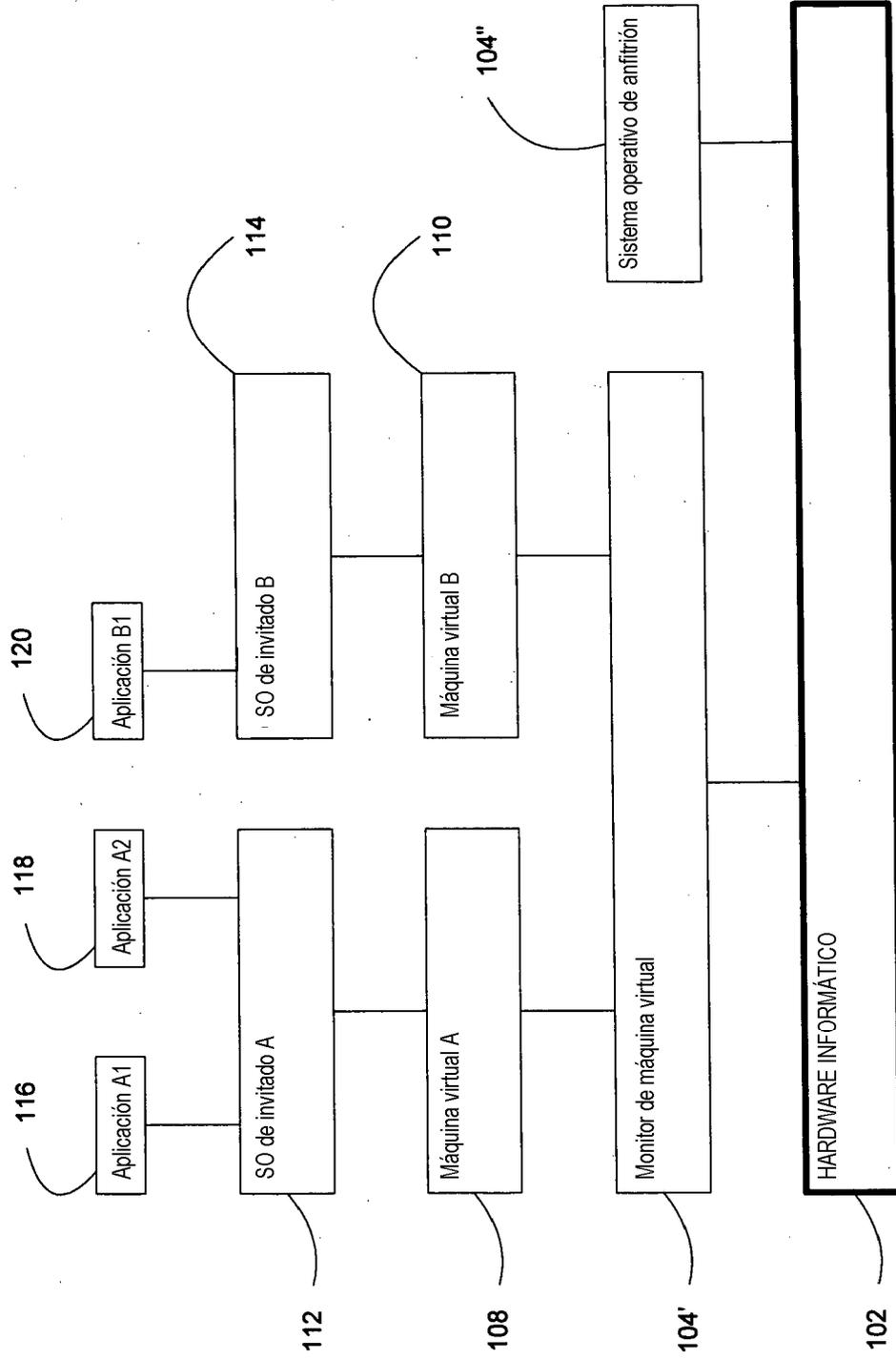


FIG. 4A

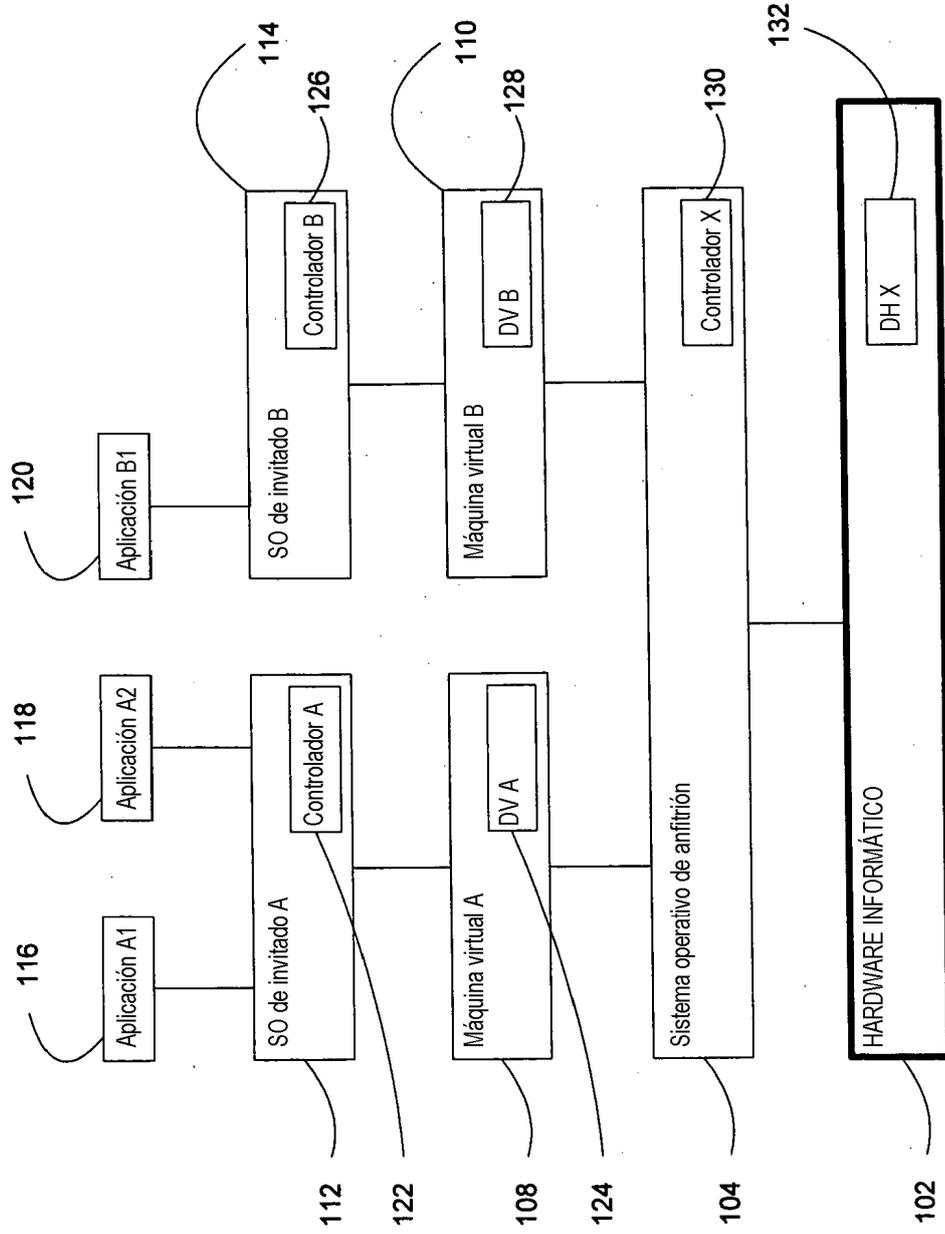


FIG. 4B

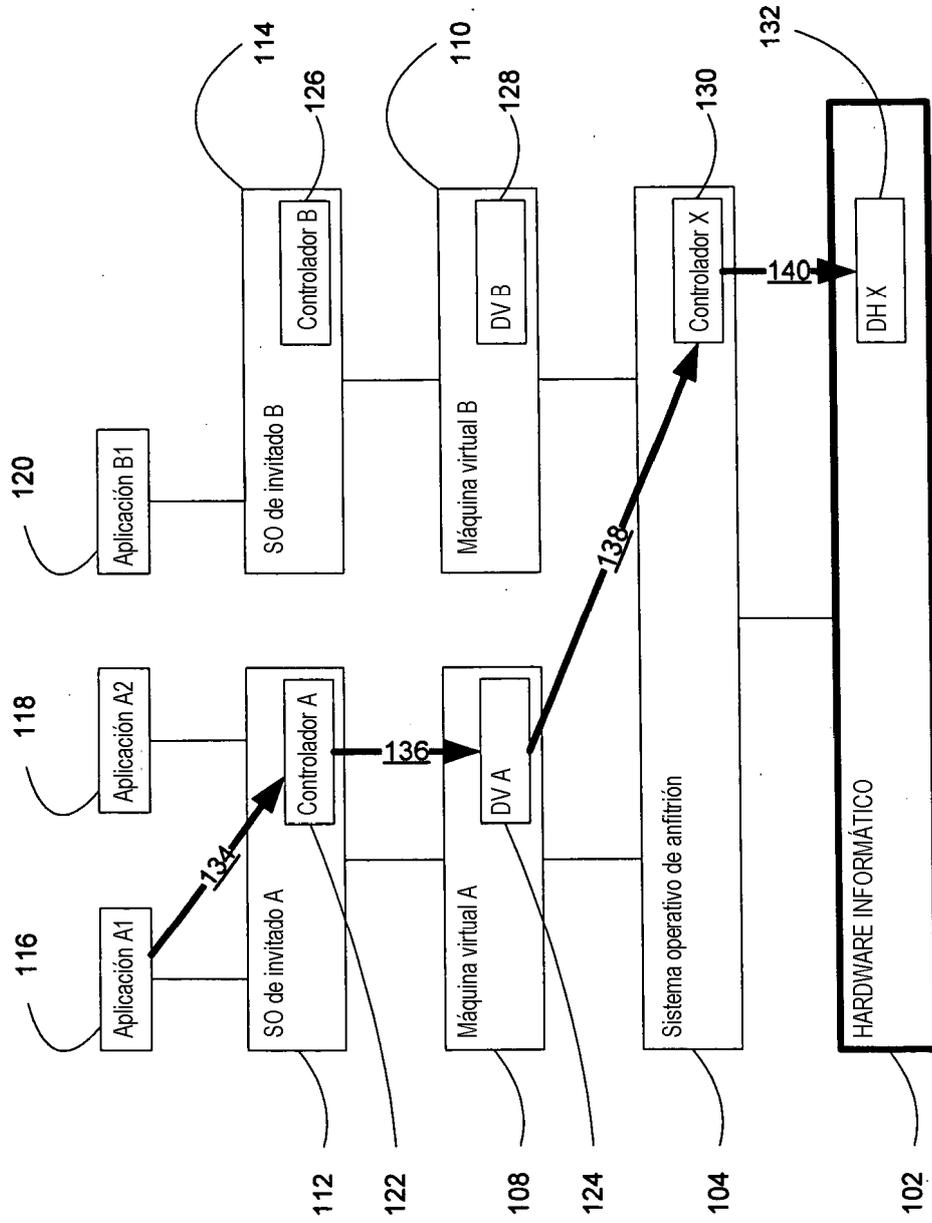


FIG. 6B

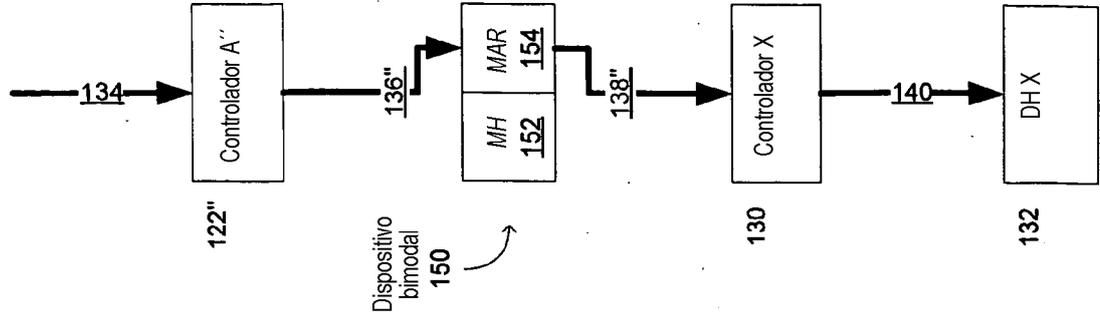


FIG. 6A

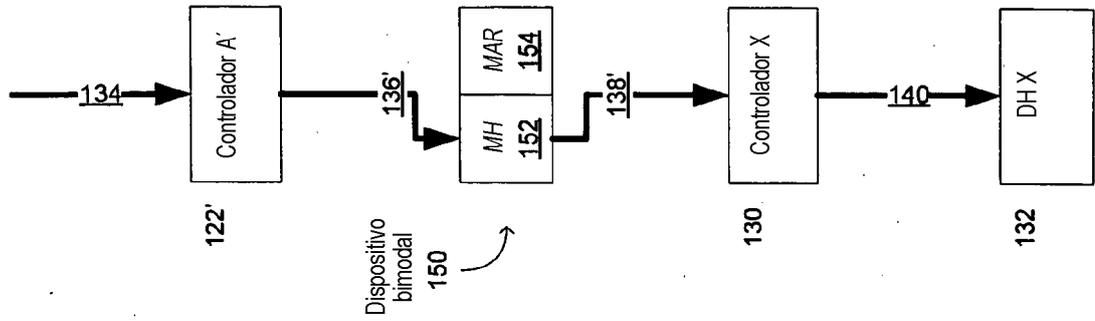


FIG. 5

