

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 382 470**

51 Int. Cl.:
G06F 9/50

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 96 Número de solicitud europea: **05111672 .1**
- 96 Fecha de presentación: **05.12.2005**
- 97 Número de publicación de la solicitud: **1674987**
- 97 Fecha de publicación de la solicitud: **28.06.2006**

54 Título: **Sistemas y procedimientos para la exposición de la topología de un procesador para máquinas virtuales**

30 Prioridad:
21.12.2004 US 18337

45 Fecha de publicación de la mención BOPI:
08.06.2012

45 Fecha de la publicación del folleto de la patente:
08.06.2012

73 Titular/es:
**MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WASHINGTON 98052-6399, US**

72 Inventor/es:
**Traut, Eric P. y
Vega, Rene Antonio**

74 Agente/Representante:
Carpintero López, Mario

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).

ES 2 382 470 T3

DESCRIPCIÓN

Sistemas y procedimientos para la exposición de la topología de un procesador para máquinas virtuales

Campo de la invención

5 La presente invención se refiere en general al campo de las máquinas virtuales (también conocidas como "virtualización de procesadores") y sistemas operativos que se ejecutan en los entornos de las máquinas virtuales. Más específicamente, la presente invención se refiere a sistemas y procedimientos para exponer la topología de un procesador de una máquina virtual a un sistema operativo huésped que se ejecuta en una máquina virtual, en el que la citada topología está basada dinámicamente en asignaciones del procesador del sistema informático anfitrión y en los recursos de memoria.

10 Antecedentes de la invención

Los ordenadores incluyen unidades de procesamiento de propósito general (CPU) anfitrionas o "procesadores" que están diseñados para ejecutar un conjunto específico de instrucciones del sistema. Un grupo de procesadores que tienen una arquitectura o especificaciones de diseño similares pueden ser considerados como miembros de la misma familia de procesadores. Ejemplos de las familias de procesadores actuales incluyen a la familia de procesadores Motorola 680X0, fabricados por Motorola, Inc. de Phoenix, Arizona, la familia de procesadores Intel 80X86, fabricados por Intel Corporation, de Sunnyvale, California, y la familia de procesadores PowerPC, que son fabricados por Motorola, Inc. y que se utilizan en los ordenadores fabricados por Apple Computer, Inc., de Cupertino, California. Aunque un grupo de procesadores puedan ser de la misma familia debido a su arquitectura y consideraciones de diseño similares, los procesadores pueden variar ampliamente dentro de una familia de acuerdo con su velocidad de reloj y otros parámetros de rendimiento.

20 Cada familia de microprocesadores ejecuta las instrucciones que son únicas para la familia de procesadores. El conjunto colectivo de instrucciones que un procesador o familia de procesadores pueden ejecutar se conoce como conjunto de instrucciones de procesador. Como ejemplo, el conjunto de instrucciones utilizado por la familia de procesadores Intel 80X86 es incompatible con el conjunto de instrucciones utilizado por la familia de procesadores PowerPC. El conjunto de instrucciones Intel 80X86 está basado en el formato Conjunto de Instrucciones Complejas para Ordenador (CISC). El conjunto de instrucciones PowerPC de Motorola está basado en el formato Conjunto de Instrucciones Reducido para Ordenador (RISC). Los procesadores CISC utilizan un gran número de instrucciones, algunas de las cuales pueden realizar funciones bastante complicadas, pero requieren, en general, muchos ciclos de reloj para ser ejecutadas. Los procesadores RISC utilizan un número menor de instrucciones disponibles para realizar un conjunto más simple de las funciones que se ejecutan a una velocidad mucho mayor.

25 La singularidad de la familia de procesadores entre los sistemas informáticos también resulta típicamente en la incompatibilidad entre los otros elementos de la arquitectura de hardware de los sistemas informáticos. Un sistema informático fabricado con un procesador de la familia de procesadores Intel 80X86 tendrá una arquitectura de hardware que es diferente de la arquitectura de hardware de un sistema informático fabricado con un procesador de la familia de procesadores PowerPC. Debido a la singularidad del conjunto de instrucciones de procesador y a la arquitectura de hardware de un sistema informático, los programas de software de aplicaciones son escritos típicamente para que se ejecuten en un sistema informático en particular que se ejecuta un sistema operativo en particular.

Máquinas Virtuales

35 Los fabricantes de ordenadores quieren maximizar su cuota de mercado teniendo más, y no menos, aplicaciones se ejecutan en la familia de microprocesadores asociados con la línea de productos del fabricante de ordenadores. Para ampliar el número de sistemas operativos y programas de aplicación que se pueden ejecutar en un sistema informático, se ha desarrollado un campo de la tecnología en el que un ordenador dado que tiene un tipo de CPU, que se denomina anfitrión, incluirá un programa de virtualización que permite que el ordenador anfitrión emule las instrucciones de un tipo no relacionado de CPU, que se denomina huésped. De esta manera, el ordenador anfitrión ejecutará una aplicación que hará que se invoquen una o más instrucciones de anfitrión como respuesta a una instrucción de huésped dada, y de esta forma, el ordenador anfitrión podrá ejecutar el software diseñado tanto para su propia arquitectura de hardware y el software escrito para ordenadores que tienen una arquitectura de hardware no relacionada.

40 Como un ejemplo más específico, un sistema informático fabricado por Apple Computer, por ejemplo, puede ejecutar sistemas operativos y programas escritos para sistemas informáticos basados en PC. También puede ser posible utilizar programas de virtualización para ejecutar concurrentemente en una sola CPU múltiples sistemas operativos incompatibles. En esta última disposición, aunque cada sistema operativo sea incompatible con el otro, los programas de virtualización pueden alojar cada uno de los distintos sistemas operativos, permitiendo así que los sistemas operativos, que de otra forma serían incompatibles, se ejecuten al mismo tiempo en el mismo sistema informático anfitrión.

55 Cuando un sistema informático huésped es emulado en un sistema informático anfitrión, se dice que el sistema informático huésped es una "máquina virtual", puesto que el sistema informático huésped sólo existe en el sistema

informático como una representación pura de software de la operación de una arquitectura de hardware específica. Las expresiones dispositivo de virtualización, emulador, ejecutor directo, máquina virtual, y emulación de procesador a veces se utilizan de manera intercambiable para indicar la capacidad de imitar o emular la arquitectura de hardware de un sistema informático completo utilizando uno o varios enfoques conocidos y apreciados por los expertos en la técnica. Por otra parte, todos los usos de la palabra "emulación" en cualquier forma pretenden transmitir este significado amplio y no se pretende distinguir entre los conceptos de ejecución de instrucciones de emulación con respecto a la ejecución directa de las instrucciones del sistema operativo en la máquina virtual. De esta manera, por ejemplo, el software Virtual PC creado por la Corporación Connectix de San Mateo, California, "emula" (por la emulación de ejecución de instrucciones y / o ejecución directa) un ordenador completo que incluye un procesador Intel Pentium 80X86 y varios componentes de la placa base y tarjetas, y la operación de estos componentes es "emulada" en la máquina virtual que se ejecuta en la máquina anfitriona. Un programa de virtualización que se ejecuta en el software del sistema operativo y arquitectura de hardware del ordenador anfitrión, tal como un sistema informático que tenga un procesador PowerPC, imita el funcionamiento de todo el sistema informático huésped.

El programa de virtualización actúa como intercambio entre la arquitectura de hardware de la máquina anfitriona y las instrucciones transmitidas por el software (por ejemplo, sistemas operativos, aplicaciones, etc.) que se ejecuta dentro del entorno emulado. Este programa de virtualización puede ser un sistema operativo anfitrión (HOS), que es un sistema operativo que se ejecuta directamente en el hardware del ordenador físico (y que puede comprender un hipervisor, que se discutirá en mayor detalle más adelante en la presente memoria descriptiva). Alternativamente, el entorno emulado también podría ser un monitor de máquina virtual (VMM) que es una capa de software que se ejecuta directamente sobre el hardware, quizás ejecutándose lado a lado y trabajando en conjunto con el sistema operativo anfitrión, y que puede virtualizar todos los recursos de la máquina anfitriona (así como algunos recursos virtuales) exponiendo las interfaces, que son las mismas que el hardware del VMM está virtualizando. Esta virtualización permite que el dispositivo de virtualización (así como el mismo sistema informático anfitrión) pase desapercibido por las capas del sistema operativo que se ejecutan sobre el mismo.

En resumen, la emulación de procesador permite que un sistema operativo huésped se ejecute en una máquina virtual creada por un dispositivo de virtualización que se ejecuta en un sistema informático anfitrión, comprendiendo el citado sistema informático anfitrión el hardware físico así como un sistema operativo anfitrión.

Topología de Procesador y de Memoria

Los programadores de sistemas operativos modernos tienen en cuenta la topología del procesador y de la memoria de la máquina para maximizar el rendimiento. Esto se hace generalmente en el inicio y, para un sistema operativo que se ejecuta en hardware físico, esto suele ser suficiente, ya que la topología de los procesadores de hardware físico permanece constante. El Sistema Operativo Windows (Windows XP, Windows 2003) y otros sistemas operativos determinan típicamente la topología del sistema en el momento de arranque de dos maneras: (a) mediante el examen de la memoria y la información de la topología de nodos de procesador en la Tabla Estática de Afinidad de Recursos (SRAT) de la BIOS y (b) mediante la lectura de datos de identificación de procesador autocontenidos (CPUID en procesadores x86 / x64) para determinar el Multithreading (multi-hilos de ejecución) Simultáneo específico (SMT, a.k.a) y topologías de múltiples núcleos.

Tal como se usa en la presente memoria descriptiva, el término "topología de procesador" pretende referirse de manera amplia a las características físicas de procesador y de la memoria asociada que, si son conocidas por un sistema operativo, podrían permitir teóricamente que un sistema operativo utilizase mejor los recursos de procesador asociados. La topología de procesador puede incluir, pero no se limita a lo siguiente: información de procesador estático, tal como SMT, de múltiples núcleos, y datos y / o información de la BIOS SRAT; información de NUMA estático, tal como disposiciones de los recursos de procesadores, memorias, y E / S, y cualquier cambio en lo anterior.

En un entorno de máquina virtual, sin embargo, mientras que la topología del procesador físico del "agente anfitrión" (el sistema operativo anfitrión, monitor de máquina virtual, y / o hipervisor) se mantiene constante, los recursos físicos asignados a un dispositivo de virtualización, y por lo tanto a la máquina virtual, pueden variar rápidamente en el tiempo, por lo que hace que las suposiciones de topología realizadas por el sistema operativo anfitrión que se ejecuta en la máquina virtual sean inexactas y por lo tanto, ineficaces.

Aunque la naturaleza dinámica de la topología puede ser mitigada usando siempre las mismas tareas de procesador físico para los procesadores virtuales o mediante la limitación de las asignaciones a un nodo específico, esto impactaría grave y negativamente sobre la capacidad del dispositivo de virtualización de hacer un uso óptimo de todos los recursos del anfitrión. Por lo tanto, lo que se necesita en la técnica es un medio para corregir la ineficiencia de una topología virtual cambiante sin afectar negativamente la capacidad de los dispositivos de virtualización de hacer un uso óptimo de todos los recursos del anfitrión.

Los procesadores de asignación y desasignación dinámicas en un sistema de procesamiento de datos con particiones lógicas se describen en el documento US 2003/0212884 A1.

Sumario de la invención

Diversas realizaciones de la presente invención se refieren a sistemas y procedimientos para hacer que un sistema operativo huésped tenga conocimiento de la topología del subconjunto de los recursos de anfitrión actualmente asignados al mismo. Para algunas de estas realizaciones, en el momento de arranque de la máquina virtual, una

5 Tabla de Afinidad de Recursos Estáticos (SRAT) será utilizada por el dispositivo de virtualización para agrupar la memoria física del huésped y los procesadores virtuales del huésped en nodos virtuales. Posteriormente, la memoria física de anfitrión detrás de un nodo virtual puede ser cambiada por el dispositivo de virtualización tal como sea necesario, y el dispositivo de virtualización proporcionará procesadores físicos adecuados para los procesadores virtuales en ese nodo.

Breve descripción de los dibujos

El sumario que antecede, así como la descripción detallada que sigue de realizaciones preferidas y ejemplos descritos para una mejor comprensión de la invención, se entenderán mejor cuando se lean en conjunto con los dibujos adjuntos. Con el propósito de ilustrar la invención, se muestran en los dibujos construcciones ejemplares de la invención; sin embargo, la invención no está limitada a los procedimientos e instrumentos específicos que se desvelan

15 en los dibujos:

La figura 1 es un diagrama de bloques que representa un sistema informático en el que se pueden incorporar los aspectos de la presente invención;

La figura 2 es un diagrama de bloques que representa la estratificación lógica de la arquitectura de hardware y software para un entorno operativo emulado en un sistema informático;

20 La figura 3A es un diagrama de bloques que representa un sistema informático virtualizado en el que la emulación es realizada por el sistema operativo anfitrión (ya sea directamente o por medio de un hipervisor);

La figura 3B es un diagrama de bloques que representa un sistema informático virtualizado alternativo en el que la emulación es realizada por medio de un monitor de máquina virtual que se ejecuta lado a lado con un sistema operativo anfitrión;

25 La figura 4 es un diagrama de bloques que ilustra un procesador de múltiples núcleos y un sistema de dos nodos de NUMA para el cual varias realizaciones de la presente invención pueden ser utilizadas;

La figura 5 es un diagrama de flujo de proceso que ilustra un procedimiento por el cual un dispositivo de virtualización proporciona la información topología dinámica del procesador para el sistema operativo huésped en una memoria de la máquina virtual, para ciertos ejemplos que se describen para una mejor comprensión de la presente invención, y

30

La figura 6 es un diagrama de bloques que ilustra un enfoque de divulgación y proporción de indicios de dos niveles para varias realizaciones de la presente invención.

Descripción detallada de realizaciones ilustrativas y ejemplos descritos para una mejor comprensión de la invención

35 La materia objeto de la invención se describe específicamente para cumplir con los requisitos legales. Sin embargo, la propia descripción no pretende limitar el alcance de esta patente. Por el contrario, el inventor ha contemplado que la materia objeto reivindicada también puede ser realizada en otras formas, para incluir diferentes pasos o combinaciones de pasos similares a los descritos en este documento, en conjunto con otras tecnologías actuales o futuras. Por otra parte, aunque el término "paso" puede ser utilizado aquí para connotar los diferentes elementos de los procedimientos empleados, el término no se debe interpretar como implicando cualquier orden particular con o entre los distintos pasos descritos en la presente memoria descriptiva, a menos que, y excepto cuando, el orden de los pasos individuales se describa explícitamente.

40

Entorno del Ordenador

45 Numerosas realizaciones de la presente invención se pueden ejecutar en un ordenador. La figura 1 y la explicación que sigue pretenden proporcionar una breve descripción general de un entorno informático adecuado en el cual la invención puede ser implementada. Aunque no se requiere, la invención se describirá en el contexto general de instrucciones ejecutables por ordenador, tales como módulos de programa, que son ejecutadas por un ordenador, tal como una estación de trabajo cliente o un servidor. En general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos y otros similares, que llevan a cabo determinadas tareas o implementan tipos de datos abstractos particulares. Además, los expertos en la técnica apreciarán que la invención puede ser practicada con otras configuraciones de sistemas informáticos, incluyendo dispositivos de mano, sistemas de múltiples procesadores, electrónica de consumo programable o basada en microprocesadores, ordenadores en red, miniordenadores, grandes ordenadores y otros similares. La invención también puede ser practicada en entornos informáticos distribuidos en los que las tareas son realizadas por los dispositivos de procesamiento remotos que

50

están enlazados a través de una red de comunicaciones. En un entorno informático distribuido, los módulos de programa pueden estar localizados en ambos dispositivos de almacenamiento de memoria local y remota.

Como se muestra en la figura 1, un sistema informático de propósito general ejemplar incluye un ordenador personal convencional 20 o similar, que incluye una unidad de procesamiento 21, una memoria de sistema 22, y un bus de sistema 23 que acopla los diversos componentes del sistema, incluyendo la memoria de sistema a la unidad de procesamiento 21. El bus de sistema 23 puede ser cualquiera de varios tipos de estructuras de bus, incluyendo un bus de memoria o controlador de memoria, un bus periférico, y un bus local utilizando cualquiera de una variedad de arquitecturas de bus. La memoria de sistema incluye memoria de sólo lectura (ROM) 24 y memoria de acceso aleatorio (RAM) 25. Un sistema básico de entrada / salida 26 (BIOS), que contiene las rutinas básicas que ayudan a transferir información entre los elementos dentro del ordenador personal 20, tal como durante el arranque, está almacenado en la ROM 24. El ordenador personal 20 puede incluir, además, un controlador de disco duro 27 para leer y escribir en un disco duro, no mostrado, un controlador de disco magnético 28 para leer o escribir en un disco removible magnético 29, y un controlador de disco óptico 30 para leer o escribir en un disco óptico removible 31 tal como un CD - ROM u otros soportes ópticos. El controlador de disco duro 27, el controlador de disco magnético 28, y el controlador de disco óptico 30 están conectados al bus de sistema 23 por una interfaz de controlador de disco duro 32, una interfaz de controlador de disco magnético 33, y una interfaz de controlador óptico 34, respectivamente. Los controladores y sus medios legibles por ordenador asociados proporcionan un almacenamiento no volátil de instrucciones, estructuras de datos, módulos de programa y otros datos legibles por ordenador para el ordenador personal 20. Aunque el entorno ejemplar descrito en la presente memoria descriptiva emplea un disco duro, un disco magnético removible 29 y un disco óptico extraíble 31, será apreciado por los expertos en la técnica que otros tipos de medios de ordenador legibles que pueden almacenar datos, que son accesible por un ordenador tales como casetes magnéticas, tarjetas de memoria flash, discos de video digital, cartuchos de Bernoulli, memorias de acceso aleatorio (RAM), memorias de sólo lectura (ROM) y otros similares, también pueden ser utilizados en el entorno operativo ejemplar.

Un número de módulos de programa puede ser almacenado en el disco duro, disco magnético 29, disco óptico 31, ROM 24 o RAM 25, incluyendo un sistema operativo 35, uno o más de programas de aplicación 36, otros módulos de programa 37 y datos de programa 38. Un usuario puede introducir comandos y la información en el ordenador personal 20 a través de dispositivos de entrada tales como un teclado 40 y un dispositivo señalador 42. Otros dispositivos de entrada (no mostrados) pueden incluir un micrófono, una palanca de mando, tableta de juegos, disco satélite, un escáner y otros similares. Estos y otros dispositivos de entrada a menudo están conectados a la unidad de procesamiento 21 por medio de una interfaz de puerto serie 46 que está acoplada al bus de sistema, pero pueden estar conectados por otras interfaces, tales como un puerto paralelo, puerto de juego o por bus serie universal (USB). Un monitor 47 u otro tipo de dispositivo de visualización está conectado también al bus de sistema 23 por medio de una interfaz, tal como un adaptador de vídeo 48. Además del monitor 47, los ordenadores personales típicamente incluyen otros dispositivos periféricos de salida (no mostrados), tales como altavoces e impresoras. El sistema ejemplar de la figura. 1 también incluye un adaptador anfitrión 55, un bus 56 de Interfaz de Sistemas Informáticos Pequeños (SCSI), y un dispositivo de almacenamiento externo 62 conectado al bus 56 de la SCSI.

El ordenador personal 20 puede operar en un entorno de red mediante conexiones lógicas a uno o más ordenadores remotos, tal como un ordenador remoto 49. El ordenador remoto 49 puede ser otro ordenador personal, un servidor, un enrutador, un PC de red, un dispositivo de pares u otro nodo de red común, y típicamente incluye muchos o todos los elementos que se han descritos más arriba en relación con el ordenador personal 20, aunque sólo un dispositivo de memoria de almacenamiento 50 se ha ilustrado en la figura. 1. Las conexiones lógicas representadas en la figura. 1 incluyen una red de área local (LAN) 51 y una red de área amplia (WAN) 52. Esos entornos de redes son muy comunes en las oficinas, redes informáticas de empresas grandes, intranets e Internet.

Cuando se utiliza en un entorno de red LAN, el ordenador personal 20 se conecta a la LAN 51 a través de una interfaz de red o adaptador 53. Cuando se utiliza en un entorno de red WAN, el ordenador personal 20 típicamente incluye un módem 54 u otro medio para establecer comunicaciones sobre la red de área amplia 52, tal como Internet. El módem 54, que puede ser interno o externo, está conectado al bus de sistema 23 a través de la interfaz de puerto serie 46. En un entorno de red, los módulos de programa representados en relación con el ordenador personal 20, o porciones de los mismos, pueden ser almacenados en el dispositivo de almacenamiento de memoria remoto. Se apreciará que las conexiones de red que se muestran son ejemplares y otros medios para establecer un enlace de comunicaciones entre los ordenadores pueden ser utilizados. Además, aunque se contempla que numerosas realizaciones de la presente invención son particularmente bien adecuadas para los sistemas informáticos, nada en este documento pretende limitar la invención a tales realizaciones.

55 Máquinas Virtuales

Desde un punto de vista conceptual, los sistemas informáticos comprenden generalmente una o más capas de software que se ejecutan en una capa fundacional de hardware. Esta estratificación se realiza por motivos de abstracción. Al definir la interfaz de una capa dada de software, esa capa puede ser implementada de forma diferente por otras capas situadas por encima de ella. En un sistema informático bien diseñado, cada capa sólo conoce (y sólo depende de) la capa inmediatamente debajo de ella. Esto permite que una capa o una "pila" (múltiples capas adyacentes) sea sustituida sin impactar negativamente en las capas situadas por encima de la citada capa o pila. Por

ejemplo, las aplicaciones de software (capas superiores) normalmente se basan en los niveles inferiores del sistema operativo (capas inferiores) para escribir archivos en alguna forma de almacenamiento permanente, y estas aplicaciones no necesitan entender la diferencia entre la escritura de datos en un disquete, en un disco duro, o en una carpeta de red. Si esta capa inferior es sustituida con nuevos componentes del sistema operativo para escribir archivos, la operación de las aplicaciones de software de capa superior no se ve afectada.

La flexibilidad del software estratificado permite que una máquina virtual (VM) presente una capa de hardware virtual que es, de hecho, otra capa de software. De este modo, una VM puede crear la ilusión a las capas de software por encima de la misma que las citadas capas de software se están ejecutando en su propio sistema informático privado, y por lo tanto, las VM pueden permitir que se ejecuten múltiples "sistemas huéspedes" de forma simultánea en un único "sistema anfitrión". Este nivel de abstracción está representado por la ilustración de la figura. 2.

La figura 2 es un diagrama que representa el estratificado lógico de la arquitectura de hardware y software para un entorno operativo emulado en un sistema informático. En la figura, un programa de emulación 94 se ejecuta directamente o indirectamente en la arquitectura de hardware física 92. El programa de emulación 94 puede ser (a) un monitor de máquina virtual que se ejecuta junto a un sistema operativo anfitrión, (b) un sistema operativo anfitrión especializado que tiene capacidades de emulación nativas, o (c) un sistema operativo anfitrión con un componente de hipervisor, en el que el citado componente de hipervisor realiza la citada emulación. El programa de emulación 94 emula una arquitectura huésped de hardware 96 (que se muestra con líneas discontinuas para ilustrar el hecho de que este componente es la "máquina virtual", es decir, hardware que en realidad no existe sino que por el contrario es emulado por el citado programa de emulación 94). Un sistema operativo huésped 98 es ejecutado en la citada arquitectura de hardware huésped 96, y la aplicación de software 100 es ejecutada en el sistema operativo huésped 98. En el sistema operativo emulado de la figura. 2, y 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 generalmente es incompatible con el sistema operativo anfitrión y con la arquitectura de hardware 92.

La figura 3A ilustra un sistema informático virtualizado que comprende un capa de software 104 del sistema operativo anfitrión que se ejecuta directamente por encima del hardware del ordenador físico 102, en el que el sistema operativo anfitrión (OS anfitrión) 104 proporciona acceso a los recursos del hardware del ordenador físico 102 mediante la exposición de interfaces que son las mismas cuando el hardware del OS anfitrión está emulando (o "virtualizando"), lo cual, a su vez, permite al OS anfitrión pasar desapercibido por las capas del sistema operativo que se ejecutan sobre el mismo. Una vez más, para realizar la emulación, el sistema operativo anfitrión 102 puede ser un sistema operativo especialmente diseñado con capacidades de emulación nativas o, alternativamente, puede ser un sistema operativo estándar con un componente hipervisor incorporado para la realización de la emulación (no mostrado).

Haciendo de nuevo a la figura. 3A, por encima del OS anfitrión 104 hay dos implementaciones de máquina virtual (VM), VM A 108, que puede ser, por ejemplo, un procesador Intel 386 virtualizado, y una VM B 110, que puede ser, por ejemplo, una versión virtualizada de uno de la familia de procesadores Motorola 680X0. Por encima de cada VM 108 y 110 se encuentran los sistemas operativos huéspedes (OS huéspedes) A 112 y B 114, respectivamente. Ejecutándose por encima del OS huésped A 112, hay dos aplicaciones, la aplicación A1 116 y la aplicación A2 118, y ejecutándose por encima del OS huésped B 114 se encuentra la aplicación B1 120.

En lo que respecta a la figura. 3A, es importante señalar que la VM A 108 y la VM B 110 (que se muestran en líneas de trazos) son representaciones de hardware de ordenador virtualizadas que sólo existen como construcciones de software y que son posibles debido a la ejecución de un o unos software de emulación especializado (s) que no sólo presentan VM A 108 y VM B 110 al OS Huésped A 112 y al OS Huésped B 114, respectivamente, sino que también realiza todos los pasos de software necesarios para que el OS Huésped A 112 y el OS Huésped B 114 interactúen indirectamente con el hardware del ordenador físico real 102.

La figura 3B ilustra un sistema informático virtualizado alternativo en el que la emulación es efectuada por un monitor de máquina virtual (VMM) 104' que se ejecuta junto con el sistema operativo anfitrión 104". En algunas realizaciones, el VMM puede ser una aplicación que se ejecuta sobre el sistema operativo anfitrión 104 e interactúa con el hardware del ordenador solamente a través del sistema operativo anfitrión 104. En otras realizaciones, y como se muestra en la figura 3B, el VMM puede comprender, por el contrario, un sistema de software parcialmente independiente que en algunos niveles interactúa indirectamente con el hardware del ordenador 102 por medio del sistema operativo anfitrión 104, pero en otros niveles, el VMM interactúa directamente con el hardware del ordenador 102 (similar a la forma en la que el sistema operativo anfitrión interactúa directamente con el hardware del ordenador). Y todavía en otras realizaciones, el VMM puede comprender un sistema de software completamente independiente que en todos los niveles interactúa directamente con el hardware del ordenador 102 (similar a la forma en la que el sistema operativo anfitrión interactúa directamente con el hardware del ordenador) sin utilizar el sistema operativo anfitrión 104 (aunque interactuando todavía con el citado sistema operativo anfitrión 104 en lo que se refiere a la coordinación del uso del citado hardware 102 y evitando conflictos y otras situaciones similares).

Todas estas variaciones para la implementación de la máquina virtual son anticipadas para formar realizaciones alternativas de la presente invención como se describe en la presente memoria descriptiva, y nada en la presente

memoria descriptiva se debe interpretar como limitante de la invención a cualquier forma de realización particular de la emulación. Además, cualquier referencia a la interacción entre las aplicaciones 116, 118, y 120 por medio de una VM A 108 y / o VM B 110, respectivamente (presumiblemente en un escenario de emulación de hardware) debe ser interpretada que es, de hecho, una interacción entre las aplicaciones 116, 118, y 120 y el dispositivo de virtualización que ha creado la virtualización. De manera similar, cualquier referencia a la interacción entre las aplicaciones de la VM A 108 y / o la VM B 110 con el sistema operativo anfitrión 104 y / o el hardware del ordenador 102 (presumiblemente para la ejecución de instrucciones de ordenador, directa o indirectamente en el hardware del ordenador 102) debe ser interpretada que es, de hecho, una interacción entre el dispositivo de virtualización que ha creado la virtualización y el sistema operativo anfitrión 104 y / o el hardware del ordenador 102, como sea apropiado.

10 Topología del Procesador

En general, un "procesador" es una circuitería lógica que responde a, y procesa las, instrucciones básicas que activan un ordenador, y también es el término que se utiliza a menudo como abreviatura de la unidad de procesamiento central (CPU). El procesador en un ordenador personal o incrustado en pequeños dispositivos es denominado a menudo microprocesador.

15 Con respecto a la topología del procesador, y como se usa en la presente memoria descriptiva, el término "procesador" se refiere específicamente a un procesador físico. Un "procesador físico" es un circuito integrado (IC) - a veces denominado "chip" o "microchip" - que comprende una oblea de semiconductor ("silicato") en la cual numerosas resistencias, condensadores, y transistores diminutos forman al menos un núcleo de procesador que comprende al menos un procesador lógico. Cada núcleo de procesador tiene la capacidad de ejecutar las instrucciones del sistema, y cada procesador lógico representa las capacidades de hyperthreading (hiper hilos de ejecución) (también conocido como multi-threading simétrico (multi-hilos de ejecución simétricos) o "SMT") mediante el cual un núcleo de procesador único aparentemente ejecuta dos hilos de ejecución en paralelo (y por lo tanto parece ser dos núcleos en el sistema).

20 Cada procesador físico está fijado en un único zócalo en una placa base de la CPU. Un procesador físico puede tener más de un núcleo de procesador (teniendo cada uno de ellos uno o más procesadores lógicos). Cada núcleo de procesador típicamente tiene su propia memoria caché de nivel 1, pero comparte una memoria caché de nivel 2 con otros núcleos de procesador en el procesador físico.

25 Un "procesador de múltiples núcleos" es un procesador físico que tiene dos o más núcleos para conseguir un mejor rendimiento, menor consumo de energía, y / o procesamiento de múltiples tareas simultáneo más eficiente (por ejemplo, procesamiento en paralelo). Por ejemplo, un "procesador de doble núcleo", que, como su nombre indica, es un procesador de múltiples núcleos que tiene dos núcleos de procesador, es algo similar a tener dos procesadores separados instalados en el mismo ordenador. Sin embargo, estos dos núcleos residen en un único procesador físico, y están conectados esencialmente en el mismo zócalo y por lo tanto, la conexión entre estos dos núcleos de procesador es más rápida de lo que sería en dos procesadores de núcleo único conectados en zócalos separados.

30 Debido a estas ganancias de rendimiento, el procesamiento con múltiples núcleos está creciendo en popularidad puesto que los procesadores de núcleo único alcanzan rápidamente los límites físicos de complejidad y velocidad posibles. Las empresas que han producido o están trabajando en productos de núcleos múltiples incluyen a AMD, ARM, Broadcom, Intel y VIA. Tanto AMD como Intel han anunciado que comercializarán los procesadores de doble núcleo en el año 2005.

35 La figura 4 es un diagrama de bloques que ilustra un procesador de múltiples núcleos y un sistema de dos nodos de NUMA para el cual se pueden utilizar varias realizaciones de la presente invención. En esta figura, un procesador físico 406 comprende dos núcleos de procesador 404 que, a su vez, comprenden dos procesadores lógicos 402. El procesador físico 406 está acoplado a la memoria 408, tal como una memoria caché L3, que es compartida y utilizada por ambos núcleos 404 de procesador físico 406. Esta figura se describe adicionalmente más adelante.

45 Topología de la Memoria

El NUMA (acceso a memoria no uniforme) es un procedimiento de configuración de un nodo de procesadores físicos en un sistema de multiprocesamiento, de manera que puedan compartir localmente la memoria, mejorando el rendimiento y la capacidad del sistema para expandirse. El NUMA se utiliza típicamente en un multiproceso simétrico (SMP) que es un sistema "ajustadamente acoplado, compartiéndolo todo" en el que múltiples procesadores que trabajan bajo un único sistema operativo acceden a la memoria de cada uno sobre un bus común o ruta "de interconexión". Por lo general, una limitación del SMP es que, a medida que se añaden microprocesadores, el bus compartido o ruta de datos se sobrecarga y se convierte en un cuello de botella del rendimiento; sin embargo, el NUMA añade un nivel intermedio de memoria (memoria del nodo) compartida entre los microprocesadores de los nodos, de manera que todos los accesos a datos no se tienen que desplazar en el bus principal.

55 Haciendo referencia de nuevo a la figura. 4, el sistema de dos nodos 416 comprende dos nodos 414, teniendo cada uno de ellos cuatro procesadores físicos 406, teniendo cada procesador físico 406 su propia memoria caché L3 compartida por los núcleos de procesador 404 de cada una de las citadas memorias caché. Además, cada procesador físico 406 y su correspondiente memoria caché L3 408 están acoplados uno al otro y a una memoria de nodo

compartido 412. Los nodos 414 y sus memorias de nodo asociadas 412 también están acoplados entre sí en este sistema de dos nodos 416, como se muestra.

Un nodo de NUMA típicamente está compuesto por cuatro procesadores físicos interconectados en un bus local a una memoria compartida (la "memoria caché L3") todo ello sobre una única placa base. Esta unidad puede ser añadida a unidades similares para formar un sistema de multiprocesamiento simétrico en el que un bus común SMP interconecta todos los nodos. El citado sistema contiene típicamente de 16 a 256 microprocesadores. Para un programa de aplicación que se ejecuta en un sistema SMP, todas las memorias de procesadores individuales parecen una memoria única.

Cuando un núcleo de procesador busca datos en una determinada dirección de memoria, en primer lugar busca en su memoria caché L1, a continuación en la memoria caché L2 de procesador físico, y a continuación en la memoria caché L3 que proporciona la configuración de NUMA antes de buscar los datos en la "memoria remota" que se encuentra localizada cerca de los otros microprocesadores. Los datos se mueven en el bus entre los grupos de un sistema NUMA SMP que utiliza una tecnología de interfaz coherente escalable (SCI). La SCI coordina lo que se llama "coherencia de memoria caché" o la consistencia entre los nodos de los grupos múltiples.

15 Exposición de la Topología del Procesador

Varias realizaciones de la presente invención se refieren a sistemas y procedimientos para la producción de un sistema operativo huésped que tienen conocimiento de la topología del subconjunto de los recursos de anfitrión actualmente asignados al mismo. Para algunas de estas realizaciones, en el momento de arranque de la máquina virtual, una Tabla de Afinidad de Recursos Estática (SRAT) será utilizada por el dispositivo de virtualización para agrupar la memoria física huésped y los procesadores virtuales huéspedes en nodos virtuales. A continuación, la memoria física anfitriona detrás de un nodo virtual puede ser cambiada por el dispositivo de virtualización como sea necesario, y el dispositivo de virtualización proporcionará procesadores físicos apropiados a los procesadores virtuales en ese nodo. Este enfoque permite sistemas operativos concedores del NUMA que se ejecutan en la máquina virtual para programar un rendimiento óptimo sin ninguna otra modificación.

Para ciertas realizaciones alternativas, el dispositivo de virtualización también puede proporcionar información de la topología dinámica del procesador para el sistema operativo huésped en la memoria de la máquina virtual. El sistema operativo huésped podría ejecutar un código adicional para recoger esta información de una posición de memoria compartida. El enfoque, conocido como "divulgación" (en el que la VM divulga información en base regular para el sistema operativo huésped y el sistema operativo huésped comprueba regularmente para obtener información actualizada y realiza el ajuste en consecuencia) requiere que el sistema operativo huésped esté provisto de un código adicional para hacer que adquiera periódicamente esta información dinámica.

La figura 5 es un diagrama de flujo del proceso que ilustra un procedimiento por el cual un dispositivo de virtualización proporciona información de la topología dinámica del procesador para el sistema operativo huésped en la memoria de la máquina virtual, en ciertos ejemplos descritos para una mejor comprensión de la presente invención. En la figura, el dispositivo de virtualización, en el paso 502, tiene conocimiento de que los recursos de hardware físicos asignados a él han cambiado. En el paso 504, el dispositivo de virtualización reconfigura la topología de procesador que está virtualizando. En el paso 506, el dispositivo de virtualización actualiza la información de la topología del procesador para el sistema operativo huésped colocando directamente la información de la topología actualizada en las tablas internas del sistema operativo huésped.

En ciertas realizaciones de la presente invención, el sistema operativo huésped ejecutaría una llamada a la máquina virtual (una llamada al dispositivo de virtualización) que designa una página de la memoria física de una máquina virtual para que sea compartida por el dispositivo de virtualización así como por el sistema operativo huésped. Esta página puede contener un campo de control con los datos de divulgación para determinar, por ejemplo: (a) si el dispositivo de virtualización debe enviar una interrupción al sistema operativo huésped cada vez que cambia la topología de la máquina virtual para que coincida con los cambios en las asignaciones de recursos del sistema informático anfitrión a la citada máquina virtual, (b) el vector que se debe utilizar para la interrupción de notificación, (c) un contador de generación que se incrementa cada vez que el hipervisor actualiza los datos de la topología, (d) una máscara de bits de todos los procesadores virtuales en el mismo procesador SMT o en el núcleo de procesador de hyperthreading, y / o (e) una máscara de bits de todos los procesadores virtuales en el mismo procesador físico, es decir, todos los procesadores lógicos en todos los núcleos en cada procesador físico. Además, los datos de la divulgación pueden abordar cualquiera de los siguientes aspectos de eficiencia: (a) prioridad de tratamientos; (b) prioridad de E / S; (c) rango de memoria protegida; (d) nodos de NUMA; (e) datos relativos al acceso a la memoria cercana y a la memoria lejana; (f) velocidad de procesador y consumo de energía del procesador; (g) zócalos y, para cada núcleo, hyperthreading; y / o (h) nivel de participación de cada procesador físico.

El programador de un sistema operativo que tiene acceso a la información de la topología dinámica del procesador y del NUMA, tal como cuando es utilizado el enfoque de "divulgación", es capaz de utilizar esta información para optimizar sus propios mecanismos de asignación de recursos (por ejemplo, la planificación de procesador, la asignación de memorias, etc.) y los esquemas de utilización de recursos. En ciertas realizaciones adicionales de la presente invención, el sistema operativo huésped (ya sea a través de llamadas al dispositivo de virtualización / máquina virtual

o a través de una página de memoria compartida) puede proporcionar indicios sobre las preferencias de asignación de recursos al dispositivo de virtualización en un proceso llamado "indicios" (que es la lógica inversa de "divulgación"). Por ejemplo, si el OS huésped preferiría tener dos procesadores virtuales asignados a dos núcleos en el mismo procesador o dos procesadores dentro del mismo nodo de NUMA por eficiencia, podría proporcionar un indicio a la VM y el programador de la máquina virtual podría tomar en cuenta estos indicios con respecto a los procesadores virtualizados, ya que se refieren a los procesadores físicos subyacentes asignados a la citada VM en cualquier momento dado. Más específicamente, estos indicios pueden abordar cualquiera de los siguientes aspectos de la eficiencia: (a) prioridad de hilos de ejecución, (b) prioridad de E / S; y / o (c) información de latencia. Por lo tanto, para las realizaciones de la presente invención, tanto el programador para el sistema operativo huésped, así como el programador para que la VM, que gestionan recursos de forma independiente, utilizan y emplean "divulgar" e "indicios" para trabajar cooperativamente para maximizar la eficiencia del sistema completo.

La figura 6 es un diagrama de bloques que ilustra un enfoque de dos niveles de divulgar y proporcionar indicios para varias realizaciones de la presente invención. En la figura, el sistema operativo huésped 602 comprende un programador 604 de OS y la máquina virtual 612 comprende un programador 614 de VM. El programador 614 de VM planifica la ejecución de los hilos de ejecución de la máquina virtual en los diversos procesadores lógicos del hardware físico a medida que tales procesadores lógicos se ponen a disposición de la máquina virtual (y que están en constante cambio), por ejemplo, por el sistema operativo anfitrión, que planifica la utilización de los citados recursos de hardware físico. Una memoria compartida 622 que se ha asignada a la máquina virtual es utilizada tanto por el programador 604 del OS huésped para proporcionar la información de "proporcionar indicios" al programador 614 de la VM, y esta memoria compartida 622 también es utilizada por el programador 614 de la VM para proporcionar la información de "divulgar" al programador 604 del OS huésped. Por ejemplo, a lo largo del flujo de datos 632, el programador 614 de la VM escribe datos reveladores 642 a la memoria compartida 622 y, a lo largo del flujo de datos 634, este dato es leído por el programador 604 del OS y es utilizado por el programador 604 del OS para un uso más eficiente de los recursos de procesadores actuales que están disponibles (y que cambian dinámicamente de vez en cuando). De manera consecuente, a lo largo del flujo de datos 636, el programador 604 del OS escribe "divulgar" datos 644, a la memoria compartida 622 y, a lo largo del flujo de datos 638, este dato es leído por el programador 614 de VM y utilizado por el programador 614 de la máquina virtual para asignar de manera más eficiente (y / o solicitar) los recursos de procesadores actuales al citado sistema operativo huésped.

Conclusión

Los diversos sistemas, procedimientos y técnicas que se han descrito en la presente memoria descriptiva pueden ser implementados con hardware o software o, en su caso, con una combinación de ambos. Por lo tanto, los procedimientos y aparatos de la presente invención, o ciertos aspectos o porciones de los mismos, pueden tomar la forma de un código de programa (es decir, instrucciones) incorporado en medios tangibles, tales como disquetes, CD - ROM, discos duros, o cualquier otro medio de almacenamiento legible por máquina, en el que, cuando el código del programa se carga y es ejecutado por una máquina tal como un ordenador, la máquina se convierte en un aparato para la práctica de la invención. En el caso de la ejecución del código de programa en ordenadores programables, el ordenador generalmente incluirá un procesador, un medio de almacenamiento legible por el procesador (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 están implementados preferentemente en un nivel de procedimiento alto o de lenguaje de programación orientado a objetos para comunicarse con un sistema informático. Sin embargo, el o los programa (s) pueden ser implementados en lenguaje ensamblador o lenguaje de máquina, si se desea. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado, y combinado con implementaciones de hardware.

Los procedimientos y aparatos de la presente invención también pueden ser realizados en forma de código de programa que es transmitido sobre algún medio de transmisión, tal como por un cableado eléctrico o cableado, por medio de fibra óptica, o por medio de cualquier otra forma de transmisión, en el que, cuando el código de programa es recibido y cargado y ejecutado por una máquina, tal como una EPROM, una matriz de puertas, un dispositivo lógico programable (PLD), un ordenador cliente, una grabadora de vídeo o similar, la máquina se convierte en un aparato para la práctica de la invención. Cuando se implementa en un procesador de propósito general, el código de programa se combina con el procesador para proporcionar un aparato único que funciona para realizar la funcionalidad de indexación de la presente invención.

Aunque la presente invención ha sido descrita en relación con las realizaciones preferidas de las diversas figuras, se debe entender que otras realizaciones similares pueden ser utilizadas o modificaciones y adiciones se pueden hacer a la realización descrita para efectuar la misma función de la presente invención sin apartarse de la misma. Por ejemplo, aunque las realizaciones ejemplares de la invención se describen en el contexto de dispositivos digitales que emulan la funcionalidad de los ordenadores personales, los expertos en la técnica reconocerán que la presente invención no está limitada a tales dispositivos digitales, y como se describe en la presente solicitud, se pueden aplicar a cualquier número de dispositivos informáticos existentes o emergentes o entornos, tales como una consola de videojuegos, ordenador de mano, ordenador portátil, etc. ya sea cableado o inalámbrico, y puede ser aplicado a cualquier número de dispositivos informáticos conectados por medio de una red de comunicaciones, e interactuando a través de la red. Además, se debe destacar que una variedad de plataformas informáticas, incluyendo sistemas operativos de dispositivos de mano y otros sistemas de interfaz de hardware / software específicos de la aplicación,

se contemplan en la presente memoria descriptiva, especialmente debido a que el número de dispositivos inalámbricos en red continúa proliferando. Por lo tanto, la presente invención no se debe limitar a cualquier forma de realización simple, sino más bien interpretarse en amplitud y alcance de acuerdo con las reivindicaciones adjuntas.

5 Finalmente, las realizaciones divulgadas descritas en la presente memoria descriptiva pueden ser adaptadas para su uso en otras arquitecturas de procesador, sistemas basados en ordenador, o virtualización de sistemas, y tales realizaciones están expresamente previstas por las divulgaciones realizadas en la presente memoria descriptiva y, por tanto, la presente invención no se debe limitar a las realizaciones específicas descritas en la presente memoria descriptiva sino que se deben interpretar más ampliamente. De manera similar, el uso de instrucciones sintéticas con propósitos distintos de la virtualización de procesador también son anticipadas por las divulgaciones realizadas en la
10 presente memoria descriptiva, y cualquier utilización de este tipo de las instrucciones sintéticas en contextos distintos al de la virtualización de procesador debe ser más ampliamente leídas en las divulgaciones realizadas en la presente memoria descriptiva.

REIVINDICACIONES

- 5 1. Un procedimiento para optimizar el rendimiento de un sistema operativo huésped (112) que se ejecute en una máquina virtual (108), y teniendo la citada máquina virtual una topología dinámica del procesador virtual basada en los cambios en un conjunto de recursos asignados dinámicamente por un sistema informático central (104), teniendo la citada máquina virtual una topología dinámica de procesador virtual, comprendiendo el citado procedimiento la actualización del citado sistema operativo huésped después del inicio con al menos una actualización para reflejar al menos un cambio en la citada topología de procesador virtual, en el que el citado paso de actualización es realizado por un dispositivo de virtualización asociado con la citada máquina virtual (108) que actualiza una posición de memoria compartida con una divulgación, y el citado sistema operativo huésped (112), accediendo posteriormente a la citada posición de memoria compartida, recuperar la divulgación y aplicar la citada divulgación,
- 10 en el que el citado sistema operativo huésped (112) proporciona un indicio acerca de las preferencias de asignación de recursos al citado dispositivo de virtualización mediante la actualización de la posición de memoria compartida con un indicio, y accediendo posteriormente el citado dispositivo de virtualización a la citada posición de memoria compartida, recuperando el indicio, y aplicando el citado indicio
- 15 en el que el sistema operativo huésped comprende un programador (604) de sistema operativo y la máquina virtual comprende un programador (614) de máquina virtual que programa la ejecución de los hilos de ejecución de la máquina virtual en varios procesadores lógicos del hardware físico puesto que tales procesadores lógicos están disponibles para la máquina virtual,
- 20 en el que la posición de memoria compartida es utilizada tanto por el programador (604) del sistema operativo huésped para proporcionar el indicio al programador (614) de la máquina virtual y esta posición de memoria compartida es utilizada también por el programador (614) de máquina virtual para proporcionar la divulgación al programador (604) del sistema operativo huésped, y
- 25 en el que el indicio es utilizado por el programador (614) de máquina virtual para asignar los recursos actuales de procesadores al citado sistema operativo huésped.
- 30 2. El procedimiento de la reivindicación 1, en el que el paso del citado sistema operativo huésped (112) de acceder posteriormente a la citada posición de memoria compartida se realiza de acuerdo con un programa o se realiza como respuesta a una interrupción generada por el citado emulador en la citada máquina virtual (108).
- 35 3. Un medio legible por ordenador que comprende instrucciones legibles por ordenador para optimizar el rendimiento de un sistema operativo huésped (112) adecuado para la ejecución en una máquina virtual (108), y teniendo la citada máquina virtual (108) una topología dinámica de procesador virtual basada en los cambios en un conjunto de recursos asignados dinámicamente por un sistema informático anfitrión (104), teniendo la citada máquina virtual una topología dinámica de procesador virtual, comprendiendo las citadas instrucciones legibles por ordenador:
- 40 instrucciones para la actualización del citado sistema operativo huésped (112) después del inicio con al menos una actualización para reflejar al menos un cambio en la citada topología de procesador virtual que comprende, además, instrucciones por lo que el citado paso de actualización se lleva a cabo por un dispositivo de virtualización asociado con la citada máquina virtual (108) que actualiza una posición de memoria compartida con una divulgación, y accediendo posteriormente el citado sistema operativo huésped (112) a la citada posición de memoria compartida, recuperando la divulgación y aplicando la citada divulgación y en el que el citado sistema operativo huésped (112) proporciona un indicio acerca de las preferencias de asignación de recursos al citado dispositivo de virtualización mediante la actualización de la posición de memoria compartida con un indicio, y accediendo posteriormente el citado dispositivo de virtualización a la citada posición de memoria compartida, recuperando el indicio, y aplicando el citado indicio,
- 45 en el que el sistema operativo huésped comprende un programador (604) de sistema operativo y la máquina virtual comprende un programador (614) de máquina virtual que programa la ejecución los hilos de ejecución de la máquina virtual en varios procesadores lógicos del hardware físico cuando tales procesadores lógicos se ponen a disposición de la máquina virtual,
- 50 en el que la posición de memoria compartida es utilizada por el programador (604) del sistema operativo huésped para proporcionar el indicio al programador (614) de máquina virtual y esta posición de memoria compartida es también utilizada por el programador (614) de máquina virtual para proporcionar la divulgación al programador (604) del sistema operativo huésped, y
- 55 en el que el indicio es utilizado por el programador de máquina virtual (614) para asignar los recursos de procesadores actuales al citado sistema operativo huésped.

4. El medio legible por ordenador de la reivindicación 3, que comprenden, además, instrucciones adicionales con lo que el paso del citado sistema operativo huésped (112) de acceder posteriormente a la citada posición de memoria compartida se realiza de acuerdo con un programa o se realiza en respuesta a una interrupción generada por el citado emulador en la citada máquina virtual (108).
5. Un sistema para optimizar el rendimiento de un sistema operativo huésped (112) que se ejecuta en una máquina virtual (108), y teniendo la citada máquina virtual una topología dinámica de procesador virtual basada en los cambios en un conjunto de recursos asignados dinámicamente por un sistema informático anfitrión (104), teniendo la citada máquina virtual una topología dinámica de procesador virtual, comprendiendo el citado sistema:
- 10 al menos un subsistema para la actualización del citado sistema operativo huésped (112) después del inicio con al menos una actualización para reflejar al menos un cambio en la citada topología de procesador virtual, y
- 15 al menos un subsistema con lo que el citado paso de actualización es realizado por un dispositivo de virtualización asociado con la citada máquina virtual (108) que actualiza una posición de memoria compartida con una divulgación, y accediendo posteriormente el citado sistema operativo huésped (112) a la citada posición de memoria compartida, recuperando la divulgación, y aplicando la citada divulgación y
- 20 en el que el citado sistema operativo huésped (112) proporciona un indicio acerca de las preferencias de asignación de recursos al citado dispositivo de virtualización mediante la actualización de la posición de memoria compartida con un indicio, y accediendo posteriormente el citado dispositivo de virtualización a la citada posición de memoria compartida, recuperando el indicio, y aplicando el citado indicio,
- 25 en el que el sistema operativo huésped comprende un programador (604) de sistema operativo y la máquina virtual comprende un programador (614) de máquina virtual que programa la ejecución de los hilos de ejecución de la máquina virtual en varios procesadores lógicos del hardware físico a medida que tales procesadores lógicos se ponen a disposición de la máquina virtual,
- 30 en el que la posición de memoria compartida es utilizada tanto por el programador (604) de sistema operativo huésped para proporcionar el indicio al programador (614) de máquina virtual y esta posición de memoria compartida es también utilizada por el programador (614) de máquina virtual para proporcionar la divulgación al programador (604) de sistema operativo huésped, y
- en el que el indicio es utilizado por el programador (614) de máquina virtual para asignar los recursos de procesadores actuales al citado sistema operativo huésped.
6. El sistema de la reivindicación 5, que comprende, además, al menos un subsistema mediante el cual el paso del citado sistema operativo huésped (112) accediendo posteriormente a la citada posición de memoria compartida se realiza de acuerdo con un programa o se realiza en respuesta a una interrupción generada por el citado emulador en la citada máquina virtual (108).
7. El sistema de las reivindicaciones 5 – 6, que comprende, además, una señal de portadora electromagnética capaz de ser transmitida a través de un medio de comunicación.

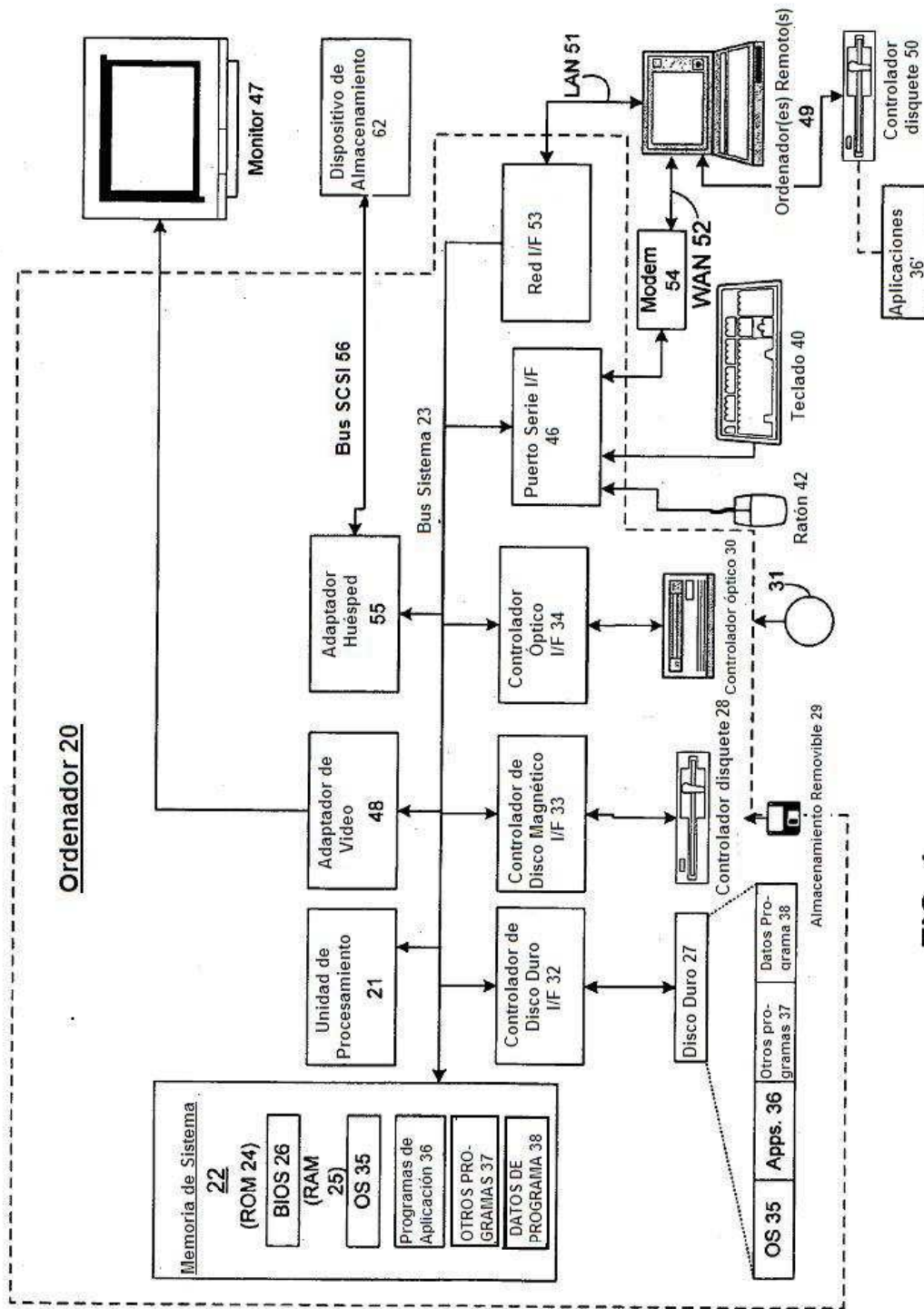


FIG. 1

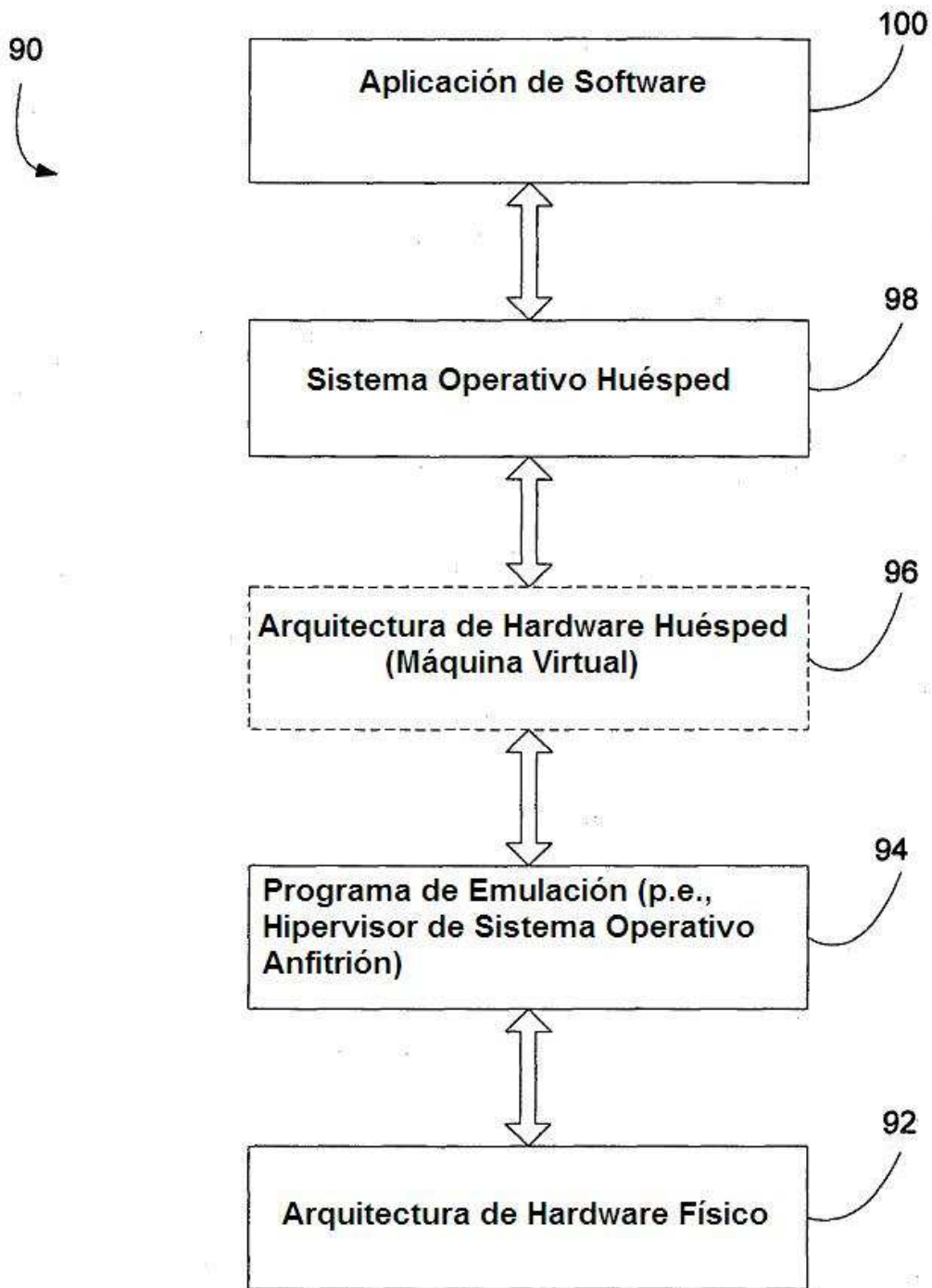


FIG. 2

FIG. 3A

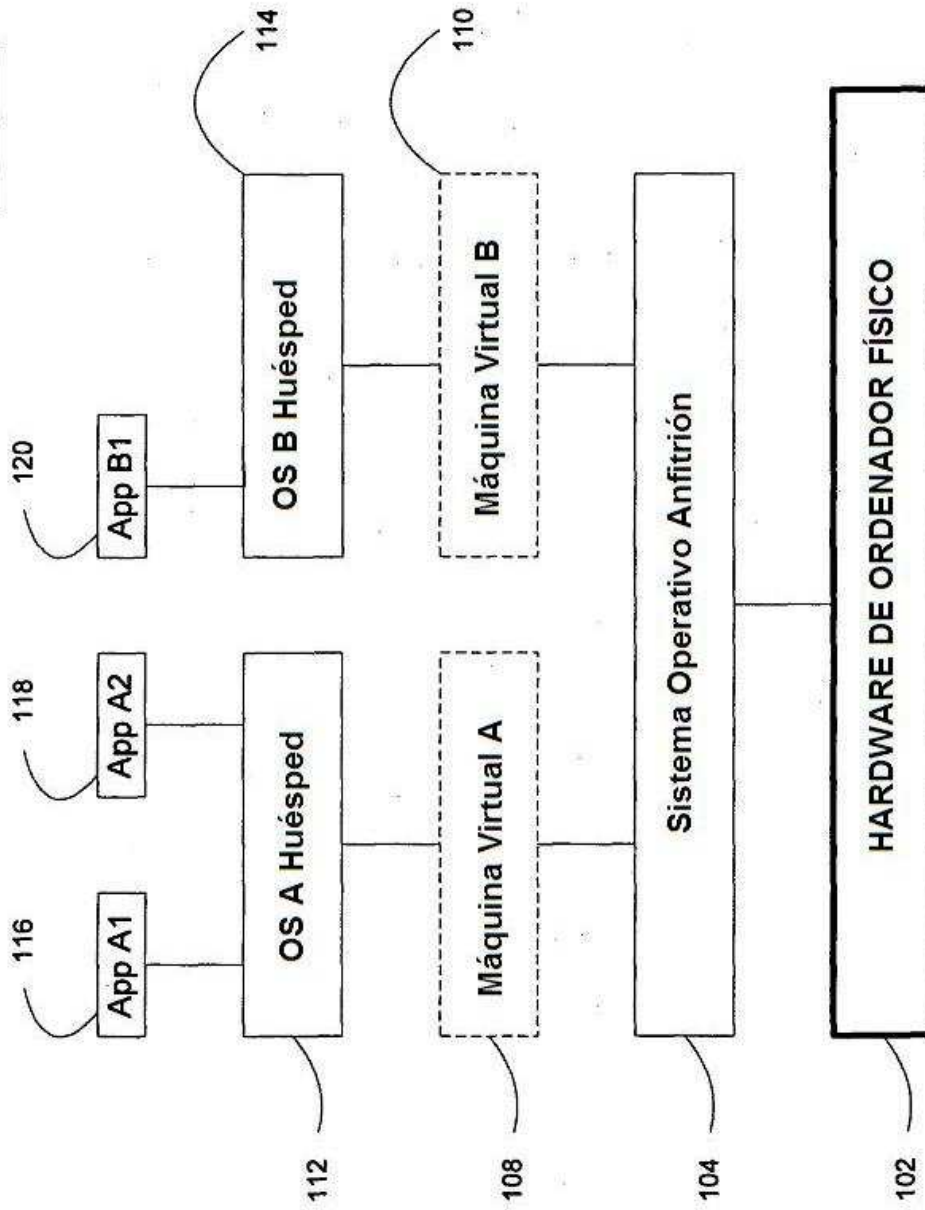
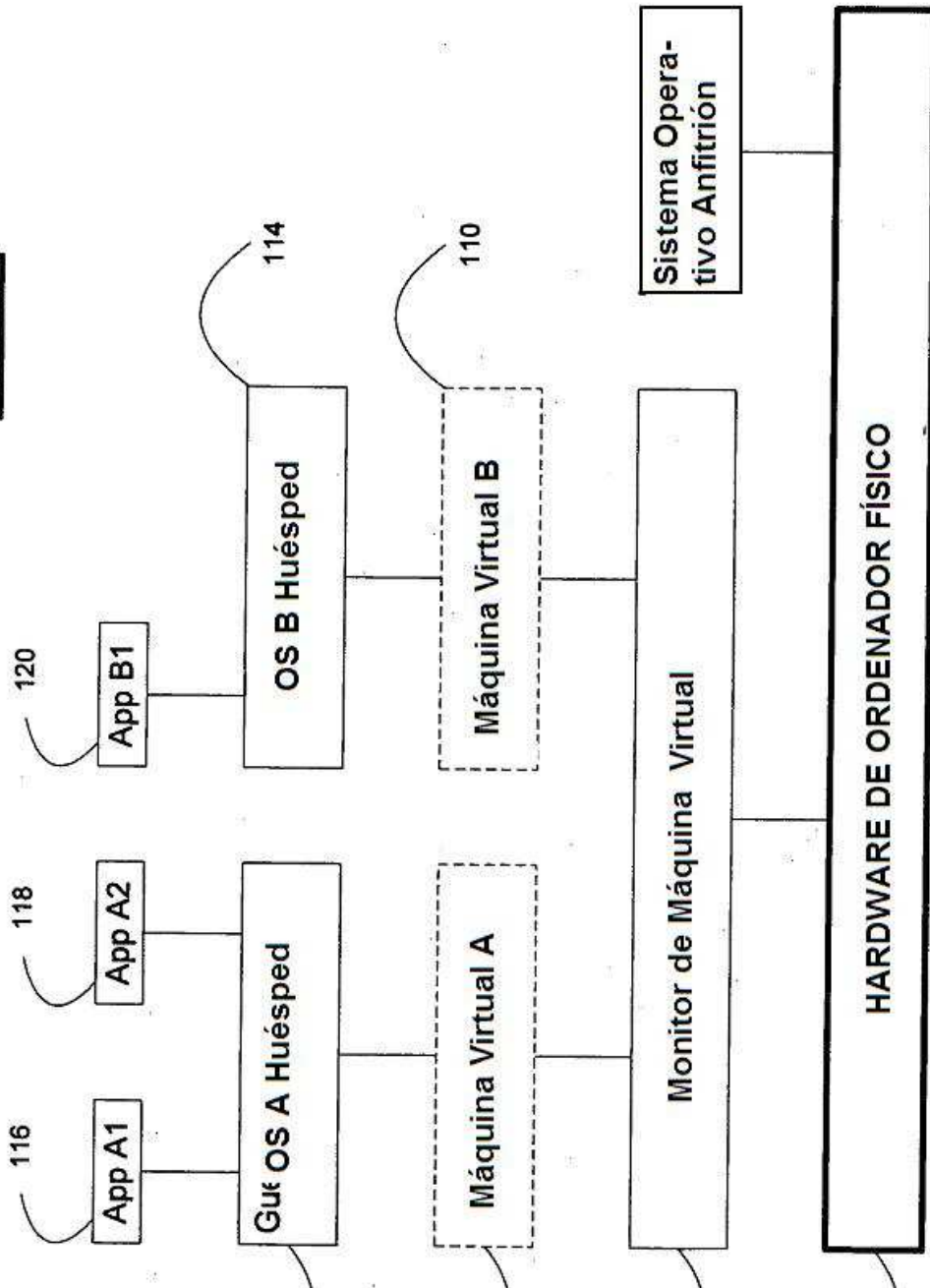


FIG. 3B



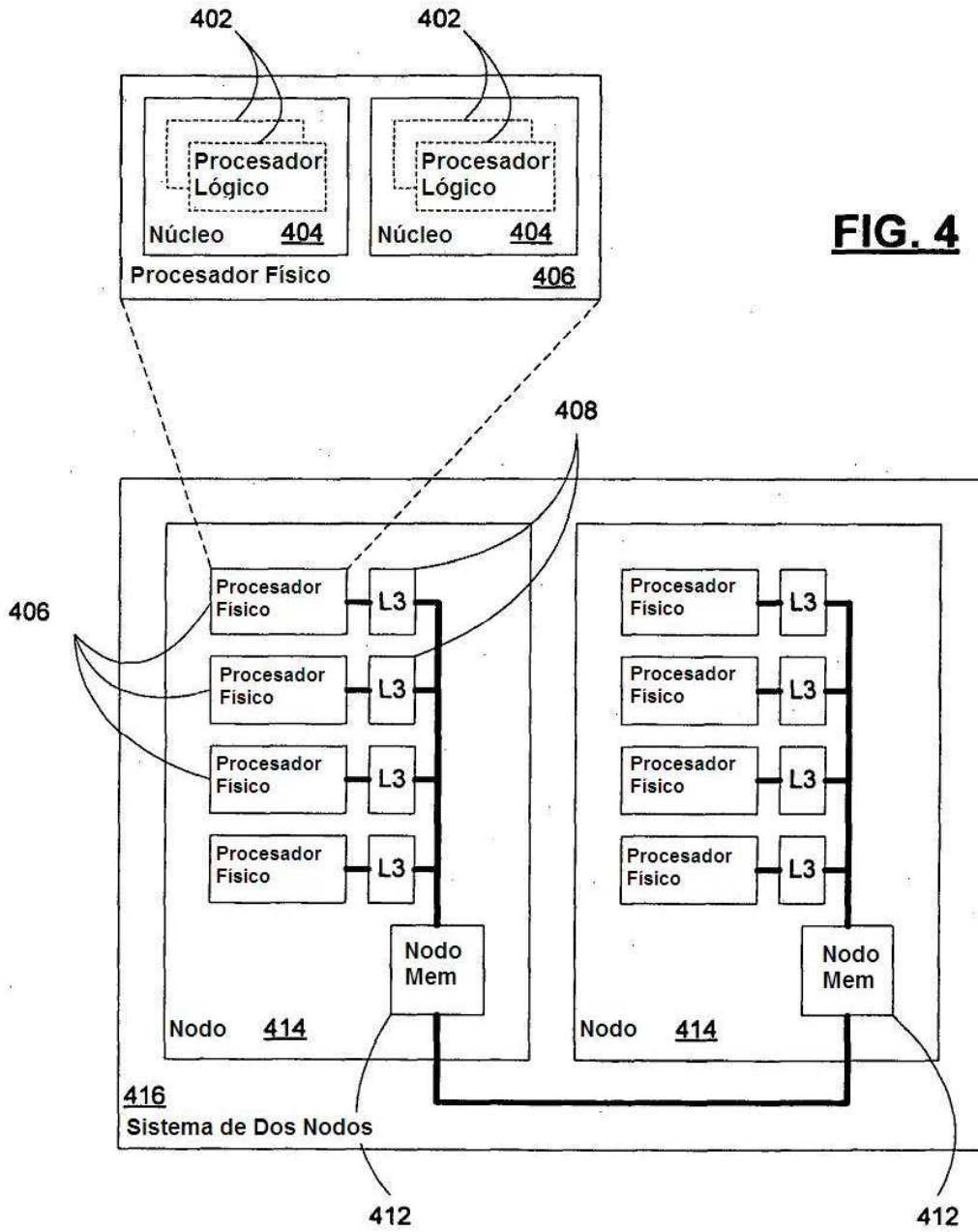


FIG. 5

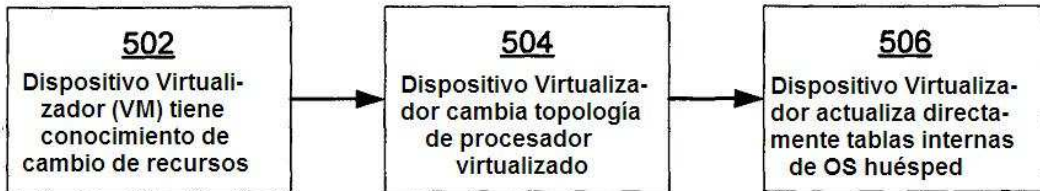


FIG. 6

