(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 734 248**

(51) Int. CI.:

G06F 9/48 (2006.01)
G06F 9/455 (2008.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **30.06.2003 E 03014873 (8)**

(97) Fecha y número de publicación de la concesión europea: **01.05.2019 EP 1380947**

(54) Título: **Procedimiento para bifurcar o migrar una máquina virtual**

(30) Prioridad:

11.07.2002 US 193531

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
05.12.2019

(73) Titular/es:

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

(72) Inventor/es:

TRAUT, ERIC P. y
VEGA. RENE A.

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Procedimiento para bifurcar o migrar una máquina virtual

Campo técnico de la invención

La presente invención se refiere en general al campo de las máquinas virtuales y, más particularmente, a un procedimiento para aplicar los conceptos de bifurcación y migración a máquinas virtuales.

Antecedentes de la invención

Los ordenadores incluyen unidades de procesamiento central (CPU) de propósito general que están diseñadas para ejecutar un conjunto específico de instrucciones del sistema. Un grupo de procesadores que tienen una arquitectura o especificaciones de diseño similares pueden considerarse miembros de la misma familia de procesadores. Algunos ejemplos de familias de procesadores actuales incluyen la familia de procesadores Motorola 680X0, fabricada por Motorola, Inc. de Phoenix, Arizona; la familia de procesadores Intel 80X86, fabricada por Intel Corporation de Sunnyvale, California; y la familia de procesadores PowerPC, que es fabricada por Motorola, Inc. y utilizada en ordenadores fabricados por Apple Computer, Inc. de Cupertino, California. Aunque un grupo de procesadores puede estar en la misma familia debido a su arquitectura y diseño similares, los procesadores pueden variar ampliamente dentro de una familia de acuerdo con la velocidad de su reloj y otros parámetros de rendimiento.

Cada familia de microprocesadores ejecuta instrucciones que son exclusivas de la familia de procesadores. El conjunto colectivo de instrucciones que puede ejecutar un procesador o una familia de procesadores se conoce como el conjunto de instrucciones del 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 se basa en el formato de ordenador con conjunto de instrucciones complejo (CISC, Complex Instruction Set Computer). El conjunto de instrucciones de Motorola PowerPC se basa en el formato de ordenador con conjunto de instrucciones reducido (RISC, Reduced Instruction Set Computer). Los procesadores CISC utilizan una gran cantidad de instrucciones, algunas de las cuales pueden realizar funciones bastante complicadas, pero que generalmente requieren muchos ciclos de reloj para ejecutarse. Los procesadores RISC utilizan una cantidad menor de instrucciones disponibles para realizar un conjunto más simple de funciones que se ejecutan a una velocidad mayor.

La singularidad de la familia de procesadores entre los sistemas informáticos también suele dar como resultado una 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 diferente a 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 del procesador y la arquitectura de hardware de un sistema informático, los programas de aplicación se escriben normalmente para ejecutarse en un sistema informático en particular que ejecuta un sistema operativo en particular.

Un fabricante de ordenadores querrá maximizar su funcionalidad haciendo que se ejecuten más aplicaciones en lugar de menos en la familia de microprocesadores asociados con la línea de productos del fabricante de ordenadores. Para ampliar la cantidad de sistemas operativos y programas de aplicación que se pueden ejecutar en un sistema informático, se ha desarrollado un campo de tecnología en el que un ordenador determinado con un tipo de CPU, denominada host, ejecutará un programa emulador que le permitirá a el ordenador host emular la recepción y ejecución de instrucciones de un tipo de CPU no relacionado, denominado invitado. Por lo tanto, el ordenador host ejecutará una aplicación que hará que se llame a una o más instrucciones de host en respuesta a una instrucción de invitado dada. En algunos casos, el ordenador host puede ejecutar software diseñado para su propia arquitectura de hardware, a excepción del programa de emulación, y software escrito para ordenadores que tienen una arquitectura de hardware no relacionada. Como 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 usar un programa emulador para operar simultáneamente varios sistemas operativos incompatibles en una sola CPU. En esta disposición, aunque cada sistema operativo es incompatible con el otro, un programa emulador puede alojar uno de los dos sistemas operativos, lo que permite que los sistemas operativos incompatibles se ejecuten simultáneamente en el mismo sistema informático.

Cuando se emula un sistema informático invitado en un sistema informático host, se dice que el sistema informático invitado es una máquina virtual, ya que el sistema informático invitado solo existe como una representación de software del funcionamiento de la arquitectura de hardware del sistema informático invitado. Los términos “emulador” y “máquina virtual” a veces se usan indistintamente para denotar la capacidad de imitar o emular la arquitectura de hardware de un sistema informático completo. Como ejemplo, el software de PC virtual creado por Connectix Corporation de San Mateo, California, emula un ordenador completa que incluye un procesador Intel

80X86 Pentium y varios componentes y tarjetas de la placa base. El funcionamiento de estos componentes se emula en la máquina virtual que se está ejecutando en la máquina host. Un programa emulador que se ejecuta en el software del sistema operativo y la arquitectura de hardware del ordenador host, como un sistema informático con un procesador PowerPC, imita el funcionamiento de todo el sistema informático invitado. El programa emulador actúa como el intercambio entre la arquitectura de hardware de la máquina host y las instrucciones transmitidas por el software que se ejecuta dentro del entorno emulado.

Una ventaja de una máquina virtual sobre una máquina real es la capacidad de crear de forma rápida y económica múltiples instancias de máquinas virtuales. Si la implementación de la máquina virtual lo permite, pueden existir múltiples máquinas virtuales simultáneamente en un entorno de una sola máquina host (sistema informático host). Los recursos de la máquina host se pueden dividir entre las diversas máquinas virtuales. Por ejemplo, una sola máquina host con cuatro procesadores y 1 gigabyte de memoria de acceso aleatorio (RAM) podría dividirse de manera uniforme en cuatro máquinas virtuales, cada una de las cuales recibe un procesador y 256 megabytes de RAM. Otras divisiones de asignación de recursos son posibles.

Esta asignación de recursos flexible se vuelve aún más útil cuando se combina con la capacidad de mover máquinas virtuales de una máquina host a otra. Esto permite el "equilibrio de carga" de sistemas. Por ejemplo, si una máquina virtual requiere más potencia de procesamiento que la que está disponible en una máquina host, se puede mover a otra máquina host que tenga capacidad adicional.

En algunos entornos informáticos, es útil tener múltiples máquinas con una configuración casi idéntica (tanto en hardware como software). Por ejemplo, un gran sitio web de comercio electrónico como Amazon.com tiene docenas o cientos de servidores web que tienen una configuración casi idéntica. Esta configuración permite una fácil expansión. Cuando la capacidad actual es inadecuada, los servidores adicionales pueden ponerse rápidamente en línea.

Otro caso en el que son útiles configuraciones casi idénticas es en la prueba de modificaciones de configuración. Cuando se trata de aplicaciones de misión crítica, los gerentes de sistemas de información a menudo desean probar los cambios de configuración del software antes de aplicarlos al sistema de producción. Por ejemplo, si Microsoft Corporation puso a disposición un nuevo "parche de seguridad" para el sistema operativo Windows, es posible que un administrador desee probar este parche en una máquina de servidor independiente antes de instalar el parche en el servidor de producción.

El documento de Kozuch, M, *et al.*: "Internet suspend/resume (Suspensión/reanudación del Internet)" Procedimientos del Cuarto Taller del IEEE sobre Sistemas y Aplicaciones Informáticos Móviles, 20-21 de junio de 2002, Piscataway, NJ, EE. UU., IEEE, páginas 40-46, ISBN: 978-0-7695-1647-9, divulga un procedimiento para combinar la tecnología de máquinas virtuales en un sistema de archivos distribuidos. Un monitor de máquina virtual encapsula todo el estado de ejecución volátil de una máquina virtual y asigna el estado volátil a los archivos en el sistema de archivos local de su host. Si estos archivos se copian a un host remoto con una arquitectura de hardware similar, el monitor de la máquina virtual en ese host puede reanudar la máquina virtual.

Sumario

El objeto de la presente invención es acelerar y simplificar un procedimiento para bifurcar o migrar una máquina virtual.

Este objeto se resuelve mediante la materia objeto de las reivindicaciones independientes.

Las realizaciones preferentes se definen por las reivindicaciones dependientes.

La presente invención en una implementación proporciona un procedimiento para aumentar la eficiencia del procesamiento de una máquina virtual. Una etapa del procedimiento es proporcionar una máquina virtual primaria. Otra etapa es suspender temporalmente la máquina virtual primaria. Otra etapa es bifurcar la máquina virtual primaria para crear una máquina virtual secundaria en una nueva ubicación.

La presente invención proporciona otro procedimiento para aumentar la eficacia del procesamiento de una máquina virtual. Una etapa del procedimiento es proporcionar una máquina virtual primaria que esté asociada con los datos almacenados. Otra etapa del procedimiento es suspender temporalmente la máquina virtual primaria. Otra etapa es bifurcar la máquina virtual primaria para crear una máquina virtual secundaria en una nueva ubicación sin al menos una primera parte de los datos almacenados.

Además, la presente invención proporciona otro procedimiento para aumentar la eficiencia del procesamiento de una máquina virtual. Una etapa del procedimiento es proporcionar una máquina virtual primaria que esté asociada con los datos almacenados. Otra etapa del procedimiento es suspender permanentemente la máquina virtual primaria. Otra etapa es migrar la máquina virtual primaria para crear una máquina virtual secundaria en una nueva

ubicación sin al menos una primera parte de los datos almacenados.

Una ventaja de una implementación de la presente invención es que hace posible la bifurcación de una máquina virtual. La capacidad de bifurcar aumenta la utilidad de las múltiples instancias de máquinas virtuales antes mencionadas. La bifurcación puede, en algunas circunstancias, crear múltiples instancias con la misma rapidez de tal forma que la mayoría de las funciones y aplicaciones que se ejecutan en las máquinas virtuales no se vean afectadas de manera significativa.

Otra ventaja de una implementación de la presente invención es que aumenta la eficiencia del procesamiento de una máquina virtual.

Una implementación del procedimiento de la presente invención tiene la ventaja de un tiempo de bifurcación, que no se ve afectado por el tamaño de la memoria de la máquina virtual primaria. Cuando la porción sin memoria de la máquina virtual primaria consta de unos pocos cientos de kilobytes de datos, el tiempo de bifurcación es de aproximadamente un milisegundo en algunos sistemas.

Si se agrega la paginación por demanda al procedimiento, entonces el tiempo total de ejecución también puede depender del tamaño de la memoria de la máquina virtual primaria. Sin embargo, la paginación por demanda no causa conflictos con las aplicaciones que se ejecutan en las máquinas virtuales primarias y secundarias.

Ninguna de las ventajas anteriores es crítica para la invención. Las implementaciones particulares de la invención pueden lograr solamente un subconjunto de las ventajas. Por ejemplo, una implementación de la invención solo puede proporcionar la opción de bifurcar una máquina virtual. Otras ventajas técnicas de la presente invención serán fácilmente evidentes para un experto en la técnica a partir de las siguientes figuras, descripciones y reivindicaciones.

Descripción detallada de la invención

En el caso de un sistema informático emulado o una máquina virtual, un programa de emulación proporciona un entorno operativo emulado en el sistema informático host. En la Figura 1 se muestra un diagrama de las capas lógicas de la arquitectura de hardware y software para un entorno operativo emulado en un sistema informático 10. Un programa de emulación 14 se ejecuta en un sistema operativo host que se ejecuta en el hardware o procesador 11 del sistema host. El programa de emulación 14 emula un sistema informático invitado 16, que incluye un sistema operativo invitado 18. Los programas de aplicación de invitado se pueden ejecutar en el sistema operativo invitado 18. En el entorno operativo emulado de la Figura 1, debido a la operación del programa de emulación 14, la aplicación invitada 20 se puede ejecutar en el sistema informático 10, aunque la aplicación invitada 20 está diseñada para ejecutarse en un sistema operativo que generalmente es incompatible con el sistema operativo host 12 y el hardware del sistema informático host 11.

Con referencia ahora a las Figuras 2 y 3, "bifurcación" es un término usado por los programadores de UNIX para describir la duplicación de un proceso de UNIX y su espacio de direcciones. Tanto el proceso original como la bifurcación se pueden ejecutar como procesos independientes desde el punto de bifurcación. La implementación de la bifurcación a menudo implica una técnica llamada "copia en escritura", en la cual todas las páginas de memoria en ambos espacios de direcciones están marcadas como "protegidas contra escritura". Cuando el proceso original o bifurcado se escribe en una página, se realiza una copia para que cada proceso tenga su propia copia. Las páginas que no se modifican pueden seguir compartiéndose entre los dos procesos. Esta técnica no solo ahorra recursos de memoria, sino que también hace que la bifurcación sea mucho más rápida de lo que sería posible.

En la presente invención, el concepto de bifurcación de un solo proceso se combina con el concepto de una máquina virtual. La presente invención permite la conversión rápida de recursos compartidos a copias privadas entre una máquina virtual original y su bifurcación. Sin embargo, el uso compartido de recursos solo es posible si ambas máquinas virtuales se ejecutan en el mismo host.

La bifurcación de la máquina virtual se puede utilizar para replicar rápidamente una máquina virtual existente. Por ejemplo, si un usuario desea probar un nuevo parche, puede bifurcar la máquina virtual y aplicar el parche a la bifurcación de no producción. Una vez que se haya probado el parche, se puede aplicar a la copia original con un riesgo limitado. Alternativamente, las máquinas virtuales de producción y no producción podrían intercambiarse una vez que se completó la prueba del parche.

En la Figura 2 se muestra un diagrama de flujo de un procedimiento 20 para bifurcar una máquina virtual. En la etapa 202, se suspende una máquina virtual primaria. En la etapa 204, se realiza una copia o "instantánea" de todas las piezas de la máquina virtual primaria, a excepción de la memoria de la máquina virtual primaria. En la etapa 206, la instantánea se mueve a una nueva ubicación, es decir, a una ubicación distinta de la ubicación de la máquina virtual primaria. Mover la instantánea a una nueva ubicación crea una nueva máquina virtual secundaria.

La máquina virtual secundaria puede o no estar ubicada en el mismo sistema de ordenador host que la máquina virtual primaria. En la etapa 208, las partes de la memoria de la máquina virtual primaria se envían a la máquina virtual secundaria utilizando la paginación por demanda. La paginación por demanda, que se muestra en la Figura 3, es un procedimiento para enviar piezas o páginas de memoria de la máquina virtual primaria a la máquina virtual secundaria. En la paginación por demanda, la memoria principal se prioriza según lo que la máquina virtual secundaria requiera activamente.

La Figura 3 es un diagrama de flujo de la paginación por demanda de la etapa 208 utilizada para bifurcar una máquina virtual. En la etapa 302, se determina si la máquina virtual primaria está a punto de modificar o no una parte de la memoria principal que aún no se ha enviado a la máquina virtual secundaria. Si la máquina virtual primaria está a punto de modificar una pieza de memoria, en la etapa 304, antes de que se le permita modificar la pieza de memoria, la pieza se envía a la máquina virtual secundaria o se hace una copia temporal de la pieza. La máquina virtual primaria guarda la copia temporal hasta un punto posterior, aún no determinado, en el procedimiento de paginación por demanda, momento en el cual la copia se envía a la máquina virtual secundaria. Si la máquina virtual primaria no está a punto de modificar una pieza de memoria o si se completa la etapa 304, el procedimiento pasa a la etapa 306.

En la etapa 306, se determina si la máquina virtual secundaria está accediendo o no a la memoria de la máquina virtual primaria. Si la máquina virtual secundaria está accediendo a la memoria de la máquina virtual primaria, en la etapa 308, la máquina virtual secundaria se suspende temporalmente y la parte de la memoria de la máquina virtual primaria requerida por la máquina virtual secundaria se envía de la máquina virtual primaria a la máquina virtual secundaria. Si la máquina virtual secundaria no está accediendo a la memoria de la máquina virtual primaria, en la etapa 310, las piezas de la memoria de la máquina virtual primaria que la máquina virtual secundaria no requiere activamente pueden enviarse de la máquina virtual primaria a la máquina virtual secundaria. Si se completa la etapa 308 o la etapa 310, el procedimiento pasa a la etapa 312.

En la etapa 312, se determina si toda la memoria de la máquina virtual primaria se ha enviado o no a la máquina virtual secundaria. Si no se ha enviado toda la memoria, el procedimiento continúa con la etapa 302, descrita anteriormente. Si se ha enviado toda la memoria, el procedimiento de paginación por demanda finaliza.

Con referencia ahora a las Figuras 4 y 5, "migrar" es un término que se refiere al movimiento de una máquina virtual de una máquina host a otra. Cuando se migra una máquina virtual, la máquina virtual original se suspende permanentemente; y la copia se ejecuta en una nueva ubicación. Una forma de implementar la migración consiste en guardar todo el estado de la máquina virtual (incluida toda su RAM) en un archivo en disco, luego copiar el archivo en el nuevo host y restaurar el estado de la máquina.

En la presente invención, la migración es similar a la bifurcación en su implementación. Al igual que con una máquina virtual bifurcada, una máquina virtual migrada puede comenzar a ejecutarse casi inmediatamente después de su creación, es decir, dentro de uno o dos segundos después de la creación. Esta característica resulta del uso de la paginación por demanda y la "copia en acceso", en la que la copia en acceso se define como copiar la memoria de la máquina virtual primaria a la máquina virtual secundaria cuando la máquina virtual secundaria accede a la memoria de la máquina virtual primaria. La copia en acceso es un aspecto de la paginación por demanda para migrar una máquina virtual.

La copia en acceso también es una ventaja de la presente invención sobre las tecnologías existentes. Las tecnologías existentes guardan y restauran una copia completa de la memoria de acceso aleatorio (RAM) de la máquina virtual primaria. El ahorro y la restauración de las tecnologías existentes pueden llevar de 5 a 60 segundos, dependiendo del tamaño de la memoria asignada a la máquina virtual primaria.

Una aplicación de migración de una máquina virtual es el equilibrio de carga. Otra aplicación sería para la conmutación por error o el mantenimiento del hardware. Por ejemplo, si el hardware en la máquina host requiere mantenimiento (por ejemplo, se agregará más memoria), la máquina virtual puede migrarse temporalmente a una máquina de respaldo, lo que evita el tiempo de inactividad.

En la Figura 4 se muestra un diagrama de flujo de un procedimiento 40 para migrar una máquina virtual. En la etapa 402, una máquina virtual primaria se suspende permanentemente. En la etapa 204, se realiza una copia o "instantánea" de todas las piezas de la máquina virtual primaria, a excepción de la memoria de la máquina virtual primaria. En la etapa 206, la instantánea se mueve a una nueva ubicación, es decir, a una ubicación distinta de la ubicación de la máquina virtual primaria. Mover la instantánea a una nueva ubicación crea una máquina virtual secundaria. La máquina virtual secundaria puede o no estar ubicada en el mismo sistema informático que la máquina virtual primaria. En la etapa 404, las piezas de la memoria de la máquina virtual primaria se envían a la máquina virtual secundaria utilizando la paginación por demanda. La paginación por demanda, representada en la Figura 5, es un procedimiento para enviar piezas o páginas de memoria de la máquina virtual primaria a la máquina virtual secundaria. En la paginación por demanda, la memoria principal se prioriza según lo que la máquina virtual secundaria requiera activamente.

La Figura 5 es un diagrama de flujo de la paginación por demanda de la etapa 404 utilizada en la migración de una máquina virtual. En la etapa 306, se determina si la máquina virtual secundaria está accediendo o no a la memoria de la máquina virtual primaria. Si la máquina virtual secundaria está accediendo a la memoria de la máquina virtual primaria, en la etapa 308, la máquina virtual de la máquina virtual secundaria se suspende temporalmente y la pieza de la memoria de la máquina virtual primaria requerida por la máquina virtual secundaria se envía de la máquina virtual primaria a la máquina virtual secundaria. Si la máquina virtual secundaria no está accediendo a la memoria de la máquina virtual primaria, en la etapa 310, las piezas de la memoria de la máquina virtual primaria que la máquina virtual secundaria no requiere activamente pueden enviarse de la máquina virtual primaria a la máquina virtual secundaria. Si se completa la etapa 308 o la etapa 310, el procedimiento pasa a la etapa 312.

En la etapa 312, se determina si toda la memoria de la máquina virtual primaria se ha enviado o no a la máquina virtual secundaria. Si no se ha enviado toda la memoria, el procedimiento continúa con la etapa 306, descrita anteriormente. Si se ha enviado toda la memoria, el procedimiento pasa a la etapa 502. En la etapa 502, se borra la memoria de la máquina virtual primaria. Después de la etapa 502, el procedimiento de paginación por demanda finaliza.

La presente invención no está limitada en su aplicación a la emulación de una arquitectura de sistema informático particular, en particular la arquitectura Intel 80X86.

Aunque la presente invención ha sido descrita en detalle, debe entenderse que se pueden realizar diversas modificaciones, sustituciones y alteraciones sin apartarse del alcance de la invención tal como se define en las reivindicaciones adjuntas.

Breve descripción de las figuras

Se puede adquirir una comprensión más completa de la presente invención y sus ventajas haciendo referencia a la siguiente descripción tomada en conjunto con las figuras adjuntas, en las que números de referencia similares indican características similares, y en las que:

La Figura 1 es un diagrama de la relación lógica de los elementos de un sistema informático emulado que se ejecuta en un sistema informático host;

La Figura 2 es un diagrama de flujo de un procedimiento para bifurcar una máquina virtual;

La Figura 3 es un diagrama de flujo de una paginación por demanda utilizada para bifurcar una máquina virtual;

La Figura 4 es un diagrama de flujo de un procedimiento para migrar una máquina virtual; y

La Figura 5 es un diagrama de flujo de una paginación por demanda utilizada en la migración de una máquina virtual.

REIVINDICACIONES

1. Un procedimiento para aumentar la eficiencia del procesamiento de máquinas virtuales, que comprende las etapas de:

proporcionar en un primer sistema host una máquina virtual primaria, en la que la máquina virtual primaria es un sistema informático emulado que es emulado por un programa emulador (14) que se ejecuta en un sistema operativo host (12), en el que el sistema operativo host se ejecuta en el primer sistema host, el programa emulador emula un sistema informático invitado (16), que incluye un sistema operativo invitado (18), en el que los programas de aplicación invitados (20) se pueden ejecutar en el sistema operativo invitado;

suspender temporalmente (202) la máquina virtual primaria;

hacer (204) una copia de todas las piezas de la máquina virtual primaria a excepción de una memoria de la máquina virtual primaria;

bifurcar la máquina virtual primaria moviendo la copia de la máquina virtual primaria a al menos una nueva ubicación para crear (206) al menos una máquina virtual secundaria y permitiendo, de este modo, el equilibrio de carga de los sistemas; y

si se determina (306) que la máquina virtual secundaria acceda a la memoria de la máquina virtual primaria, suspender temporalmente (308) la máquina virtual secundaria y enviar una parte de la memoria de la máquina virtual primaria requerida por la máquina virtual secundaria desde la máquina virtual primaria a la máquina virtual secundaria.

2. El procedimiento según la reivindicación 1, en el que la al menos una nueva ubicación está en el primer sistema informático host.

3. El procedimiento según la reivindicación 1, en el que la al menos una nueva ubicación está en un segundo sistema informático host.

4. El procedimiento según la reivindicación 1, en el que las etapas de suspender temporalmente (202) la máquina virtual primaria y bifurcar la máquina virtual primaria se ejecutan en una duración de aproximadamente un milisegundo.

5. El procedimiento según la reivindicación 1, en el que la máquina virtual primaria es un servidor de red.

6. El procedimiento según la reivindicación 1, que además comprende la etapa de:
utilizar (208) la paginación por demanda para enviar una subporción de la memoria a la máquina virtual secundaria.

7. El procedimiento según la reivindicación 6, en el que la etapa de utilizar (208) paginación por demanda además comprende la etapa de:

si no toda la memoria asociada con la máquina virtual primaria ha sido enviada a la al menos una máquina virtual secundaria, enviar una segunda subporción de la primera porción a la al menos una máquina virtual secundaria.

8. El procedimiento según la reivindicación 6, en el que la etapa de utilizar (208) paginación por demanda comprende la etapa de:
si la al menos una máquina virtual secundaria no está accediendo a la al menos una memoria de la máquina virtual primaria, enviar (310) páginas de memoria de la al menos una máquina virtual secundaria.

9. El procedimiento según la reivindicación 1, en el que la primera porción son todas las piezas de la máquina virtual primaria.

10. El procedimiento según la reivindicación 6, en el que la etapa de utilizar (208) paginación por demanda comprende la etapa de:
si la máquina virtual primaria intenta modificar una primera subporción, enviar la primera subporción a la al menos una máquina virtual secundaria antes de que la primera subporción sea modificada por la máquina virtual primaria.

11. El procedimiento según la reivindicación 10, en el que la etapa de utilizar (208) paginación por demanda además comprende la etapa de:
si no toda la primera porción de la memoria de la máquina virtual primaria ha sido enviada a la al menos una

máquina virtual secundaria, enviar una segunda subporción de la primera porción a la al menos una máquina virtual secundaria.

12. El procedimiento según la reivindicación 6, en el que la etapa de utilizar (208) paginación por demanda comprende la etapa de:
si la máquina virtual primaria intenta modificar una primera subporción, realizar una copia de la primera subporción antes de que la máquina virtual primaria modifique la primera subporción.

13. El procedimiento según la reivindicación 12, en el que la copia de la primera subporción es una copia temporal.

14. El procedimiento según la reivindicación 12, en el que la etapa de utilizar (208) paginación por demanda además comprende la etapa de:

si no toda la primera porción de la memoria de la máquina virtual primaria ha sido enviada a la al menos una máquina virtual secundaria, enviar una segunda subporción de la primera porción a la al menos una máquina virtual secundaria.

15. Un procedimiento para aumentar la eficiencia del procesamiento de máquinas virtuales, que comprende las etapas de:

proporcionar en un primer sistema host una máquina virtual primaria, en la que la máquina virtual primaria es un sistema informático emulado que es emulado por un programa emulador (14) que se ejecuta en un sistema operativo host (12), en el que el sistema operativo host se ejecuta en el primer sistema host, el programa emulador emula un sistema informático invitado (16), que incluye un sistema operativo invitado (18), en el que los programas de aplicación invitados (20) se pueden ejecutar en el sistema operativo invitado;

suspender permanentemente (402) la máquina virtual primaria;

hacer (204) una copia de todas las piezas de la máquina virtual primaria a excepción de una memoria de la máquina virtual primaria;

migrar la máquina virtual primaria moviendo la copia de la máquina virtual primaria a al menos una nueva ubicación para crear (206) al menos una máquina virtual secundaria y, por lo tanto, permitir el equilibrio de carga de los sistemas; y

si se determina (306) que la máquina virtual secundaria accede a la memoria de la máquina virtual primaria, suspender temporalmente (308) la máquina virtual secundaria y enviar una parte de la memoria de la máquina virtual primaria requerida por la máquina virtual secundaria desde la máquina virtual primaria a la máquina virtual secundaria.

16. El procedimiento según la reivindicación 15, en el que la al menos una nueva ubicación está en el primer sistema informático host.

17. El procedimiento según la reivindicación 15, en el que la al menos una nueva ubicación está en un segundo sistema informático host.

18. El procedimiento según la reivindicación 15, en el que las etapas de suspender de forma permanente (402) la máquina virtual primaria y migrar la máquina virtual primaria se ejecutan en una duración de aproximadamente un milisegundo.

19. El procedimiento según la reivindicación 15, en el que la máquina virtual primaria es un servidor de red.

20. El procedimiento según la reivindicación 15, que además comprende la etapa de:
borrar la máquina virtual primaria.

21. El procedimiento según la reivindicación 15, que además comprende la etapa de:
utilizar (404) paginación por demanda para enviar una primera porción de la memoria a la al menos una máquina virtual secundaria.

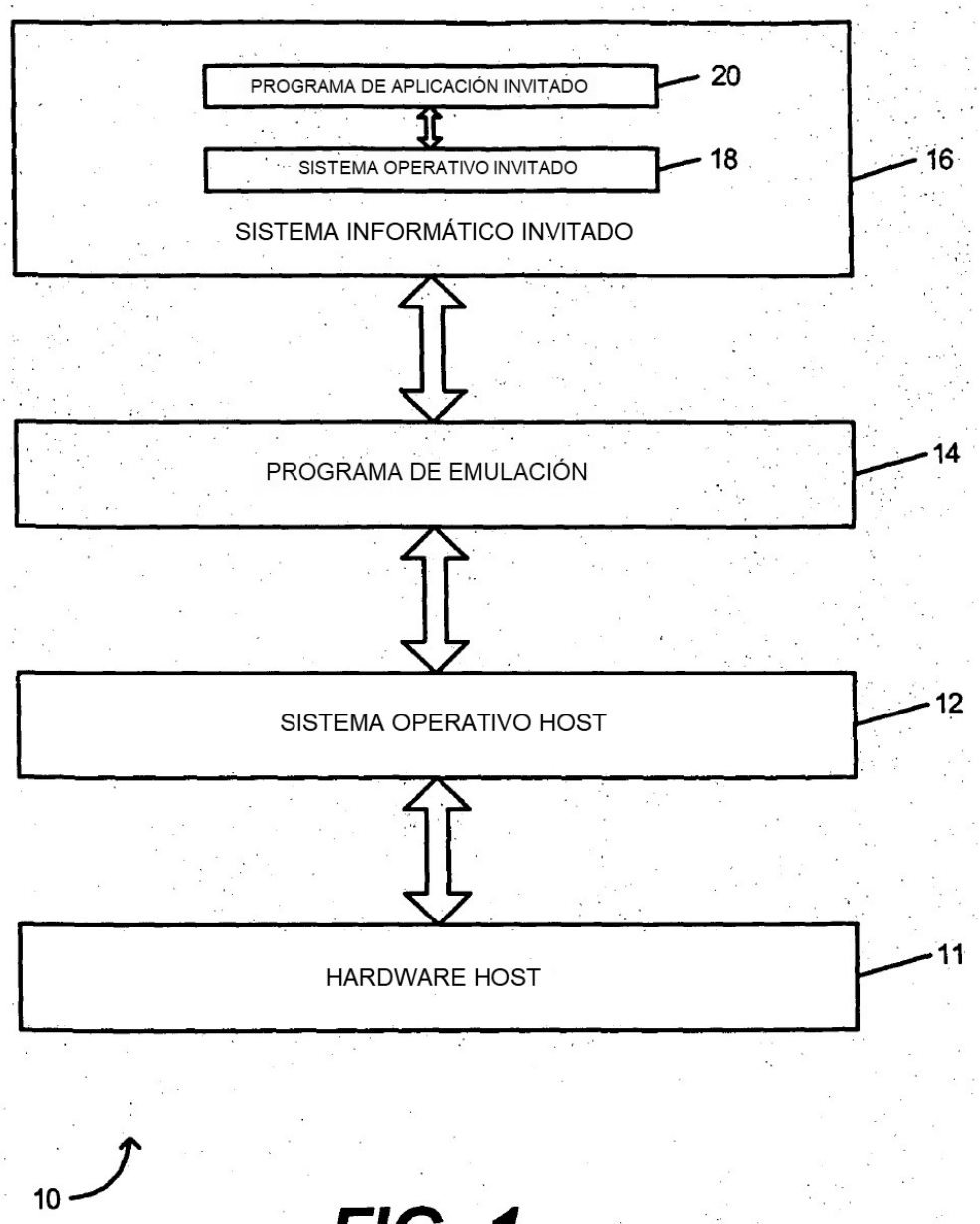
PROGRAMA DE APLICACIÓN INVITADO
20
SISTEMA OPERATIVO INVITADO
18
16
SISTEMA INFORMÁTICO INVITADO
PROGRAMA DE EMULACIÓN
14
SISTEMA OPERATIVO HOST
12
HARDWARE HOST
11
10

FIG. 1

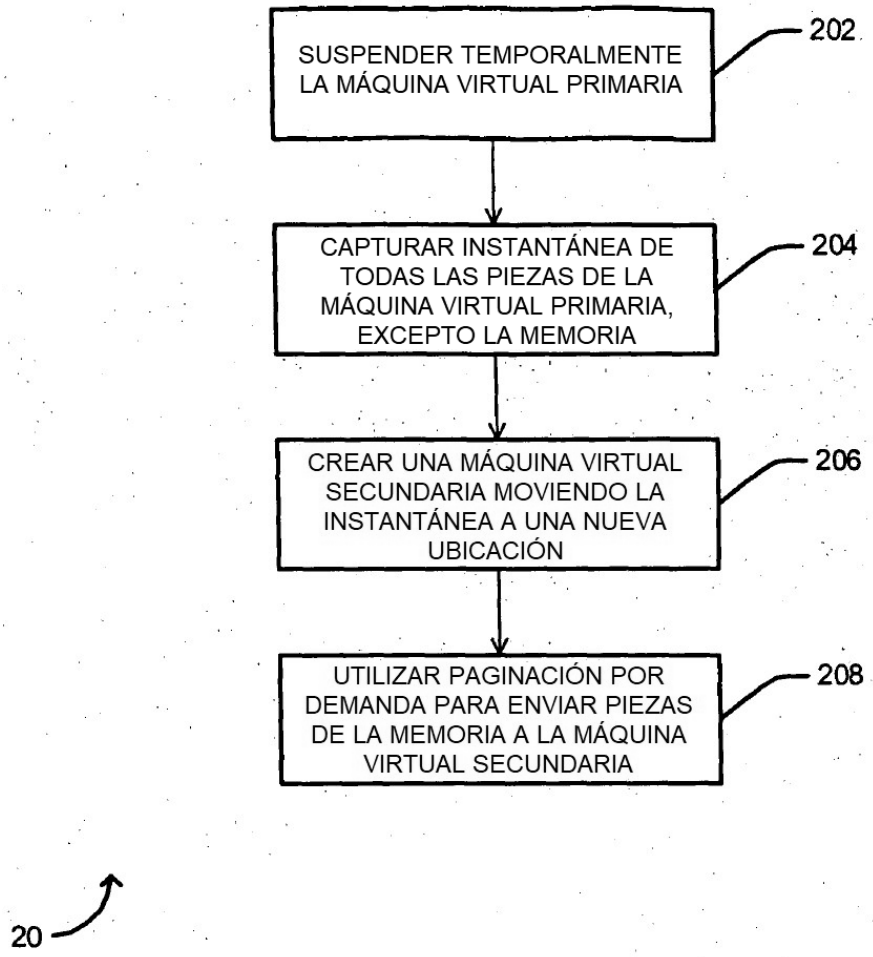
SUSPENDER TEMPORALMENTE LA MÁQUINA VIRTUAL PRIMARIA
202
CAPTURAR INSTANTÁNEA DE TODAS LAS PIEZAS DE LA MÁQUINA VIRTUAL PRIMARIA, EXCEPTO LA MEMORIA
204
CREAR UNA MÁQUINA VIRTUAL SECUNDARIA MOVIENDO LA INSTANTÁNEA A UNA NUEVA UBICACIÓN
206
UTILIZAR PAGINACIÓN POR DEMANDA PARA ENVIAR PIEZAS DE LA MEMORIA A LA MÁQUINA VIRTUAL SECUNDARIA
208
20

FIG. 2

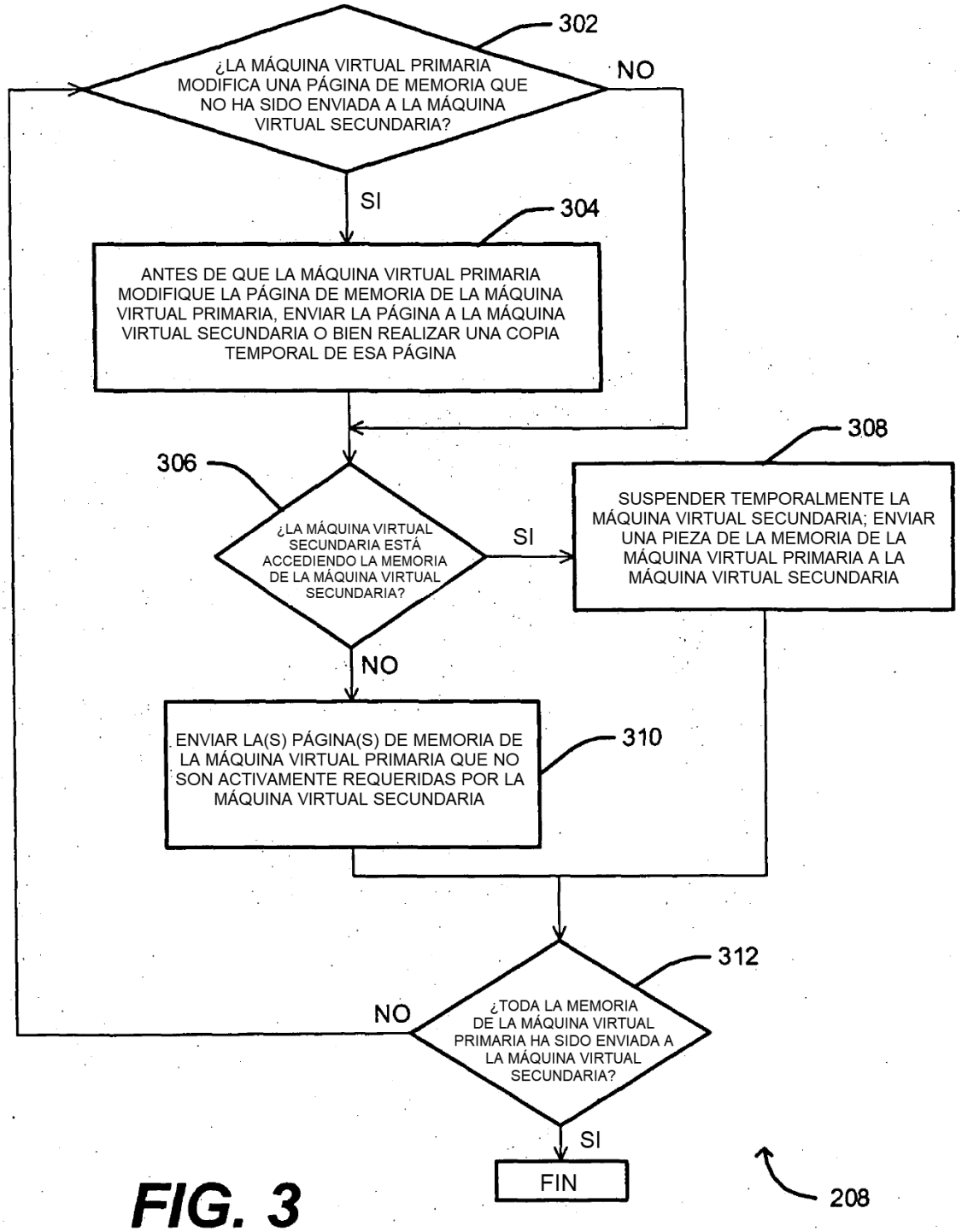
302
¿LA MÁQUINA VIRTUAL PRIMARIA MODIFICA UNA PÁGINA DE MEMORIA QUE NO HA SIDO ENVIADA A LA MÁQUINA VIRTUAL SECUNDARIA?
NO
SI
304
ANTES DE QUE LA MÁQUINA VIRTUAL PRIMARIA MODIFIQUE LA PÁGINA DE MEMORIA DE LA MÁQUINA VIRTUAL PRIMARIA, ENVIAR LA PÁGINA A LA MÁQUINA VIRTUAL SECUNDARIA O BIEN REALIZAR UNA COPIA TEMPORAL DE ESA PÁGINA
306
¿LA MÁQUINA VIRTUAL SECUNDARIA ESTÁ ACCEDIENDO LA MEMORIA DE LA MÁQUINA VIRTUAL SECUNDARIA?
SI
308
SUSPENDER TEMPORALMENTE LA MÁQUINA VIRTUAL SECUNDARIA; ENVIAR UNA PIEZA DE LA MEMORIA DE LA MÁQUINA VIRTUAL PRIMARIA A LA MÁQUINA VIRTUAL SECUNDARIA
NO
ENVIAR LA(S) PÁGINA(S) DE MEMORIA DE LA MÁQUINA VIRTUAL PRIMARIA QUE NO SON ACTIVAMENTE REQUERIDAS POR LA MÁQUINA VIRTUAL SECUNDARIA
310
312
¿TODA LA MEMORIA DE LA MÁQUINA VIRTUAL PRIMARIA HA SIDO ENVIADA A LA MÁQUINA VIRTUAL SECUNDARIA?
NO
SI
FIN
208

FIG. 3

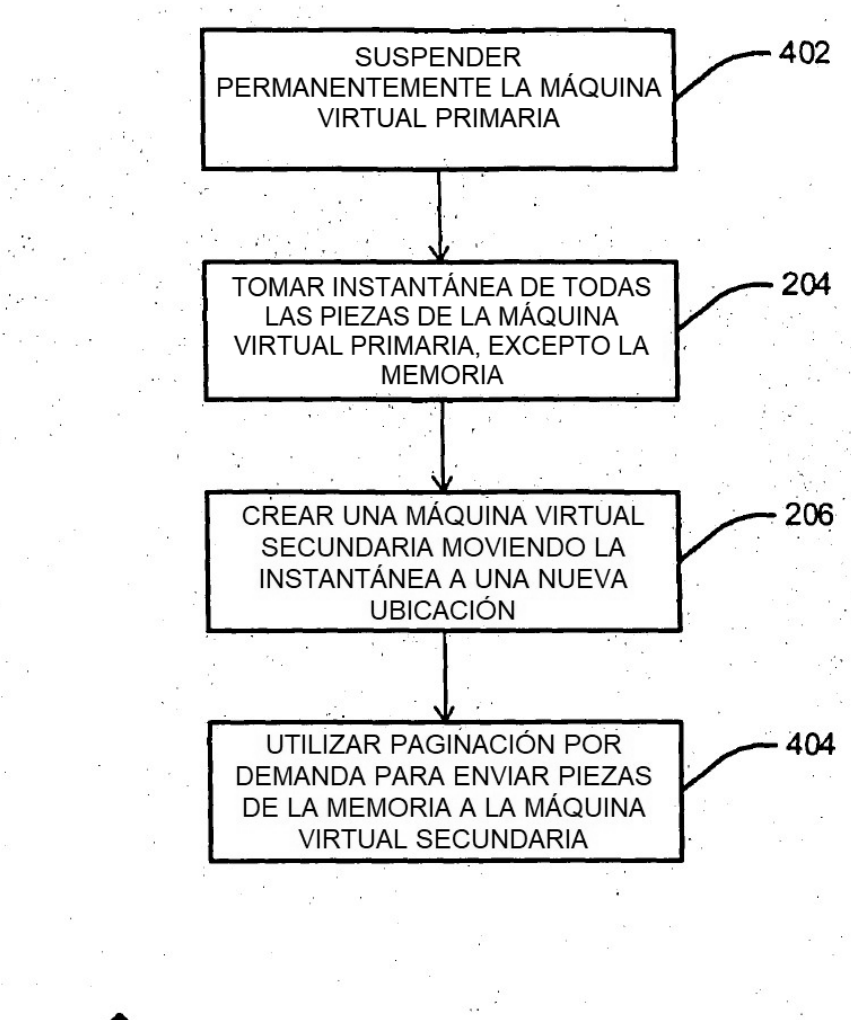
SUSPENDER PERMANENTEMENTE LA MÁQUINA VIRTUAL PRIMARIA
402
TOMAR INSTANTÁNEA DE TODAS LAS PIEZAS DE LA MÁQUINA VIRTUAL PRIMARIA, EXCEPTO LA MEMORIA
204
CREAR UNA MÁQUINA VIRTUAL SECUNDARIA MOVIENDO LA INSTANTÁNEA A UNA NUEVA UBICACIÓN
206
UTILIZAR PAGINACIÓN POR DEMANDA PARA ENVIAR PIEZAS DE LA MEMORIA A LA MÁQUINA VIRTUAL SECUNDARIA
404
40

FIG. 4

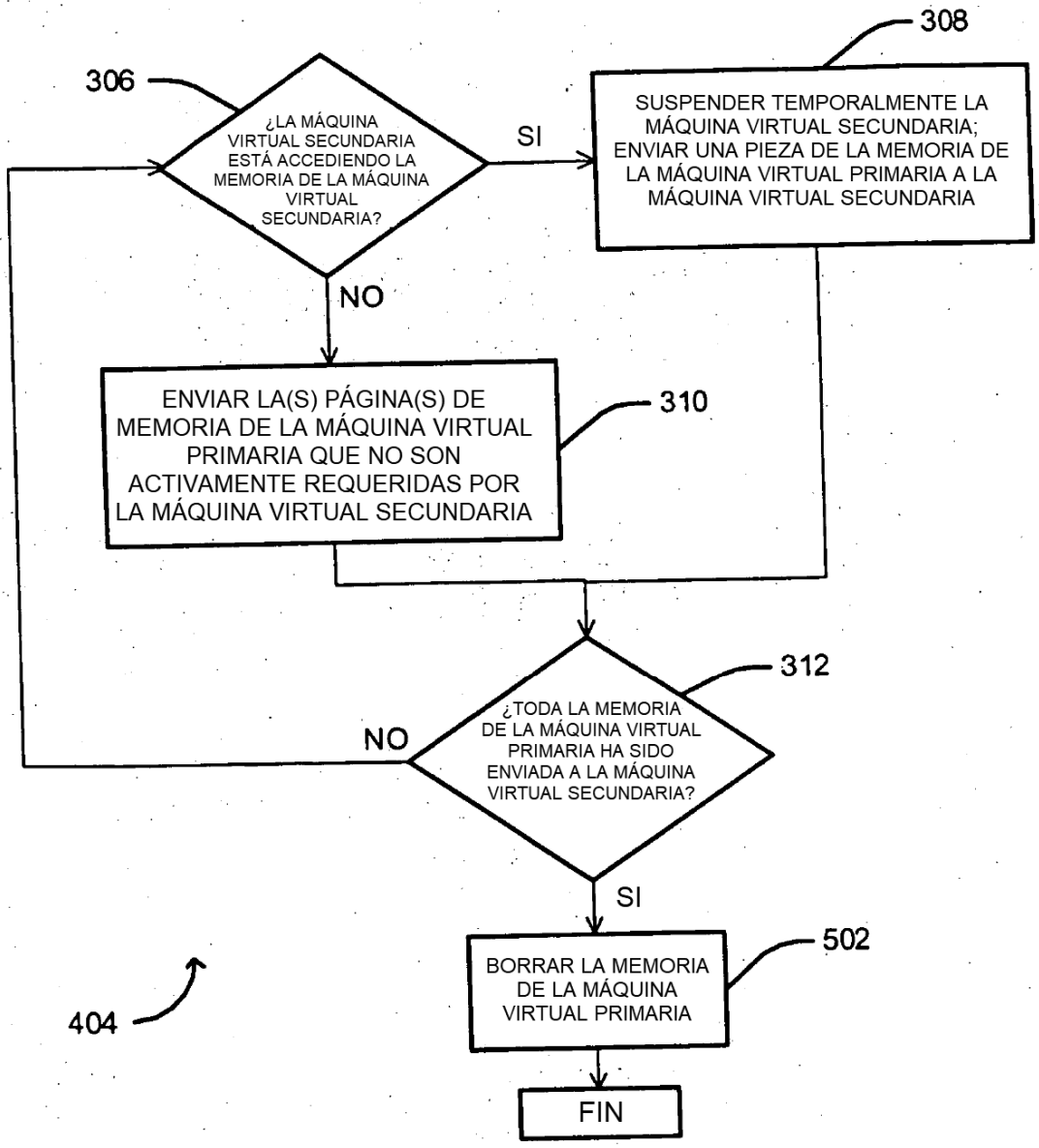
306
¿LA MÁQUINA VIRTUAL SECUNDARIA ESTÁ ACCEDIENDO LA MEMORIA DE LA MÁQUINA VIRTUAL SECUNDARIA?
SI
308
SUSPENDER TEMPORALMENTE LA MÁQUINA VIRTUAL SECUNDARIA; ENVIAR UNA PIEZA DE LA MEMORIA DE LA MÁQUINA VIRTUAL PRIMARIA A LA MÁQUINA VIRTUAL SECUNDARIA
NO
ENVIAR LA(S) PÁGINA(S) DE MEMORIA DE LA MÁQUINA VIRTUAL PRIMARIA QUE NO SON ACTIVAMENTE REQUERIDAS POR LA MÁQUINA VIRTUAL SECUNDARIA
310
312
¿TODA LA MEMORIA DE LA MÁQUINA VIRTUAL PRIMARIA HA SIDO ENVIADA A LA MÁQUINA VIRTUAL SECUNDARIA?
NO
SI
502
BORRAR LA MEMORIA DE LA MÁQUINA VIRTUAL PRIMARIA
404
FIN

FIG. 5