

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 711 561**

51 Int. Cl.:

G06F 12/08 (2006.01)
G06F 9/44 (2008.01)
G06F 12/02 (2006.01)
G06F 9/46 (2006.01)
G06F 9/50 (2006.01)
G06F 12/06 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **19.02.2016 PCT/US2016/018559**

87 Fecha y número de publicación internacional: **09.09.2016 WO16140808**

96 Fecha de presentación y número de la solicitud europea: **19.02.2016 E 16708857 (4)**

97 Fecha y número de publicación de la concesión europea: **14.11.2018 EP 3265918**

54 Título: **Uso de compresión de memoria para reducir la carga de compromiso de memoria**

30 Prioridad:

02.03.2015 US 201514635302

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

06.05.2019

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)
One Microsoft Way
Redmond, Washington 98052-6399 , US**

72 Inventor/es:

**BAK, YEVGENIY M.;
IYIGUN, MEHMET;
WANG, LANDY y
KISHAN, ARUN U.**

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 711 561 T3

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

DESCRIPCIÓN

Uso de compresión de memoria para reducir la carga de compromiso de memoria

Antecedentes

5 A medida que la tecnología de los ordenadores ha avanzado el rendimiento de los procesadores del ordenador ha aumentado. Este mayor rendimiento ha dado como resultado unos programas de ordenador más complejos que proporcionan una mayor funcionalidad a los usuarios. No obstante, este mayor rendimiento no está exento de problemas. Uno de tales problemas es que los programas de ordenador se hacen más complejos ya que usan mayores cantidades de memoria del ordenador. Esto ha llevado a la necesidad de incluir más memoria del ordenador en los ordenadores, lo que aumenta el coste de tales ordenadores. El documento US6298422 describe un método para reducir las necesidades de memoria, y el consumo del ciclo del CPU de un programa de ejecución en un estado suspendido suspende el programa interceptando los puntos de entrada del programa. Los métodos, sistemas y programas de ordenador del documento US2013/0160011 gestionan la memoria de un entorno de tiempo de ejecución en una máquina virtual. El documento US2005/0132161 describe un método de creación de un espacio de memoria virtual en una memoria.

Compendio

La presente invención está definida en las reivindicaciones independientes anejas a las que se debería hacer referencia.

Unas características ventajosas se exponen en las reivindicaciones independientes anejas.

20 Las realizaciones o ejemplos de la siguiente descripción que no están cubiertas por las reivindicaciones son consideradas como que no son parte de la invención de acuerdo con esta descripción. Este Resumen se proporciona para introducir una selección de conceptos en una forma simplificada que están además descritos más adelante en la Descripción Detallada. Este Resumen no pretende identificar características clave o características esenciales de la materia objeto reivindicada, ni se pretende que sea usado para limitar el alcance de la materia objeto reivindicada.

25 Se ha proporcionado un método de acuerdo con la reivindicación independiente 1 y un dispositivo de ordenador de acuerdo con la reivindicación independiente 10. De acuerdo con uno o más aspectos, se hace una determinación de que un programa está en un estado a partir del cual el programa puede ser terminado, una cantidad de memoria que ha sido comprometida al programa, siendo la cantidad de memoria un compromiso de memoria para el programa, y un subconjunto de la cantidad de memoria comprometida al programa que ha sido usada por el programa. La cantidad de memoria comprometida al programa incluye muchas páginas de memoria. El contenido de las páginas de memoria en el subconjunto de memoria usado por el programa es comprimido, y se determina una cantidad del compromiso de memoria para el programa que puede ser liberada. Después de que el subconjunto de memoria usado por el programa ha sido comprimido, la cantidad determinada de la memoria comprometida para el programa es liberada.

35 De acuerdo con uno o más aspectos se hace una determinación de que se va a acceder a una o más páginas de un programa, habiendo el programa tenido liberado al menos parte de su compromiso de memoria. Se realiza un intento para obtener de nuevo el compromiso de memoria para al menos una o más páginas de memoria del programa. En respuesta al intento de obtener de nuevo el compromiso de memoria que tiene éxito, se permite acceder a una o más páginas de memoria. En respuesta al intento de obtener de nuevo el compromiso de memoria sin éxito, el programa es terminado y lanzado de nuevo.

Breve descripción de los dibujos

En los dibujos se usan los mismos números como referencia de similares características.

La Figura 1 ilustra un sistema de ejemplo que emplea el uso de la compresión de memoria para reducir la carga de compromiso de memoria de acuerdo con una o más realizaciones.

45 Las Figuras 2A y 2B ilustran unos diagramas de estado de ejemplo que muestran diferentes estados en los que un programa puede estar de acuerdo con una o más realizaciones.

La Figura 3 es un diagrama de flujos que ilustra un proceso de ejemplo para conservar el compromiso de memoria cuando un programa es terminable de acuerdo con una o más realizaciones.

50 La Figura 4 es un diagrama de flujos que ilustra un proceso de ejemplo para permitir el acceso a la memoria de un programa que ha sido suspendido o de otro modo hecho terminable de acuerdo con una o más realizaciones.

La Figura 5 ilustra un sistema de ejemplo que incluye un dispositivo de cálculo de ejemplo que es representativo de uno o más sistemas y/o dispositivos que pueden aplicar las diversas técnicas aquí descritas.

Descripción detallada

El uso de la compresión de memoria para reducir la carga de compromiso de memoria (en adelante también referida como “carga de compromiso” o “compromiso de memoria”) se discute aquí. Una memoria en un dispositivo de cálculo está formado por muchos bloques o porciones denominados páginas de memoria (o simplemente páginas). Estas páginas están respaldadas por un archivo de páginas en un dispositivo de almacenamiento (por ejemplo, un disco), y un gestor de la memoria del dispositivo de cálculo gestiona la transferencia de datos de las diversas páginas entre la memoria y el archivo de páginas según sea conveniente. El gestor de la memoria también asigna cantidades de memoria a programas que funcionan en el dispositivo de cálculo.

Cuando la memoria es asignada a un programa, un espacio de direcciones es asignado o destinado al programa. Este espacio de direcciones incluye la cantidad de memoria asignada, y el gestor de la memoria destina una carga de compromiso para la cantidad de memoria asignada. Esta cantidad de memoria asignada es la cantidad de memoria que está comprometida al programa, y también se denomina como el compromiso de memoria para el programa (o simplemente el compromiso para el programa). El compromiso de memoria garantiza la disponibilidad de la cantidad de memoria asignada al programa – el gestor de la memoria garantiza que si se accede a una dirección virtual comprometida en la memoria asignada al programa, unas páginas en memoria física (por ejemplo RAM) pueden ser hechas disponibles para la dirección. En el momento en el que la memoria es asignada al programa, no es necesario que páginas físicas sean (pero alternativamente pueden ser) asignadas al programa. Una cantidad máxima de carga de compromiso permisible en el dispositivo de cálculo es igual a la cantidad de memoria física (por ejemplo RAM) más el tamaño de todos los archivos de páginas en un dispositivo de almacenamiento que respalda la memoria física (por ejemplo RAM).

El contenido de una o más de las páginas de memoria de un programa es comprimido. Esta compresión puede ocurrir mientras que el programa está activo (por ejemplo, ejecutándose o de otro modo está permitido acceder a la memoria asignada al programa) y/o en respuesta al programa que es suspendido (o de otro modo colocado en un estado no activo). Cuando un programa es suspendido, se determina un compromiso de cantidad de memoria para el programa que puede ser liberada. Esta cantidad de compromiso de memoria que puede ser liberada es la cantidad que fue comprometida al programa menos cualquier cantidad aún almacenada (en formato comprimida) de información (por ejemplo, datos o instrucciones) para el programa. La cantidad determinada de compromiso de memoria es liberada, permitiendo que el compromiso de cantidad de memoria liberada sea consumido por otros programas según convenga.

Cuando deben ser accedidas una o más páginas de memoria de un programa suspendido, tal como debido a que el programa suspendido que es hecho activo de nuevo o código de modo nuclear que se adjunta al espacio de memoria del programa suspendido, se hace un intento de adquirir de nuevo el compromiso de memoria para esas páginas de memoria. Dependiendo de qué memoria ha sido asignada a qué otros programas, el intento puede tener éxito o no tenerlo. Si el intento tiene éxito, entonces está permitido el acceso a una o más páginas de memoria. No obstante, si el intento no tiene éxito, entonces el programa es terminado, y el programa es opcionalmente lanzado de nuevo. El lanzamiento de nuevo del programa permite que el programa se ejecute, asignando una memoria como convenga al programa (y opcionalmente resultando en uno o más de otros programas que son terminados). Si el programa es lanzado de nuevo puede variar basado en el tipo de acceso (por ejemplo, el programa puede ser lanzado de nuevo si el acceso es debido a que el programa suspendido está siendo hecho activo nuevamente, pero no puede ser lanzado de nuevo si el acceso es debido al código de modo nuclear que se adjunta al espacio de memoria del programa suspendido).

La Figura 1 ilustra un sistema 100 de ejemplo que emplea el uso de la compresión de memoria para reducir la carga de compromiso de memoria de acuerdo con una o más realizaciones. El sistema 100 es aplicado por un dispositivo de cálculo e incluye un gestor 104 de memoria, un dispositivo de almacenamiento 106, y una memoria paginada 108. Generalmente, el gestor 104 de la memoria gestiona el almacenamiento de datos en la memoria paginada 108. El gestor 104 de la memoria incluye un controlador 110 de la memoria y un asignador 112 de memoria. El asignador 112 de memoria destina unas porciones de la memoria paginada 108 a diversos programas 114 en el sistema 100. La porción de la memoria paginada 108 asignada a un programa 114 también se denomina el espacio de direcciones del programa 114. El controlador 110 de la memoria controla la transferencia de páginas de memoria entre la memoria paginada 108 y el dispositivo de almacenamiento 106, así como la compresión de las diferentes porciones de la memoria paginada 108.

Un programa 114, cuando está ejecutándose, es también denominado aquí como un proceso (que es un caso de un programa 114 que está ejecutándose), de este modo porciones de la memoria paginada 108 destinadas a un programa 114 también se les conoce como asignadas a un proceso del programa 114. Las páginas asignadas a un proceso son propiedad de o dedicadas a ese proceso y son usadas por ese proceso y no otro proceso (y por tanto son también denominadas como páginas privadas). Los programas 114 pueden ser aplicaciones, programas del sistema operativo u otros componentes o módulos. El gestor 104 de la memoria recibe solicitudes desde estos programas 114 para recuperar datos desde la memoria paginada 108 y para escribir datos en la memoria paginada 108. La memoria paginada 108 puede ser, por ejemplo, cualquier tipo de memoria direccionable de la CPU (Unidad de Procesamiento Central), tal como una memoria volátil (por ejemplo, RAM) o una memoria no volátil (por ejemplo, una memoria flash).

El controlador 110 de la memoria gestiona la memoria paginada 108 usando el paginamiento. El controlador 110 de la memoria organiza la memoria 108 (por ejemplo, RAM) en páginas, que son una unidad particular (por ejemplo, fija) de tamaño de datos. La acción de paginar se refiere a leer datos en unidades de páginas del archivo de respaldo (también denominado como el archivo de intercambio), que es el archivo de páginas 116 en el sistema 100, cuando los datos no están en la memoria 108. La acción de paginar también se refiere a escribir datos sucios (modificados) de nuevo en unidades de páginas en el archivo de páginas 116. Las páginas de memoria son así también denominadas páginas de memoria respaldadas por páginas de archivo. Tales técnicas de paginamiento son bien conocidas por los expertos en la técnica.

En respuesta a una solicitud de asignación de memoria para un programa, el asignador 112 de la memoria asigna una carga de compromiso para la cantidad de memoria solicitada, que garantiza que si se tuviera que acceder a una dirección virtual en esa memoria en algún momento, las páginas físicas en la memoria serían hechas disponibles al programa. La solicitud es típicamente recibida desde el programa propiamente dicho. El contenido (por ejemplo, datos) de páginas de la memoria asignada al programa puede ser transferido a y desde el archivo de páginas 116 en momentos diferentes, pero esa transferencia es gestionada por el controlador 110 de la memoria y el programa tiene acceso a la memoria que es destinada al programa.

Alternativamente, la asignación 112 de la memoria y los programas 114 siguen un enfoque de compromiso de memoria retrasada. En el enfoque de compromiso de memoria retrasada no sucede una asignación de compromiso real cuando una solicitud de asignación de memoria es hecha por el programa 114. En vez de ello, un compromiso es asignado a petición cuando el programa 114 escribe primero en la memoria asignada. Si el compromiso no puede ser asignado en ese momento, surge una excepción y/o el programa es terminado.

El sistema 100 incluye un módulo 118 de preparación de la terminación del programa y un módulo 120 de acceso al programa. El módulo 118 de preparación de la terminación del programa gestiona poner el programa en un estado en el que el programa pueda ser terminado (también denominado como un estado terminable). Esto puede implicar la comunicación con el programa de modo que el programa guarde un estado de procesamiento del programa, que se refiere a los diversos datos, ajustes, u otra información usada por el programa. Guardando el estado de procesamiento del programa, si el programa es terminado el programa puede ser posteriormente lanzado de nuevo y devuelto al mismo estado de procesamiento como lo había tenido antes de ser terminado. Poner un programa en un estado en el que el programa puede ser terminado incluye también liberar al menos parte del compromiso de memoria para el programa, como se discute más adelante con más detalle. En una o más realizaciones un programa que es puesto en un estado en el que el programa puede ser terminado es suspendido, lo que se refiere al programa que no funciona en el sistema 100 (por ejemplo, el programa no está programado para ejecución por cualesquiera procesadores del sistema 100). Alternativamente, un programa puede ser puesto en un estado en el que el programa pueda ser terminado sin suspender el programa.

En una o más realizaciones el sistema 100 sigue un enfoque de suspensión de liberación del compromiso. En el enfoque de suspensión de liberación del compromiso el módulo 118 de preparación de la terminación del programa gestiona poner el programa en un estado en el que el programa puede ser terminado como se ha discutido antes.

Alternativamente, el sistema 100, como reivindicado, puede seguir un enfoque de liberación de tiempo de ejecución. En el enfoque de liberación de tiempo de ejecución un programa es considerado terminable en todo momento, incluso cuando el programa está activo (por ejemplo, ejecutándose). El compromiso de memoria puede ser guardado en el tiempo de ejecución cuando la aplicación está todavía activa (pero terminable) de modo que la carga de compromiso para el subconjunto de páginas en el espacio de direcciones que están comprimidas puede ser liberada tan pronto como las páginas son comprimidas en las páginas de memoria comprimida. En este enfoque (por ejemplo, usado en conjunción con el enfoque de compromiso de memoria retrasada antes discutido), el almacenamiento de los datos comprimidos (las páginas de memoria comprimida) carga su propio compromiso. Cuando el programa accede a una página liberada de compromiso, el compromiso es vuelto a cargar, una página física es asignada, y los datos son descomprimidos desde el almacén comprimido en la página física. Si la carga de compromiso falla (la carga de compromiso no puede ser adquirida de nuevo), entonces surge una excepción (y/o el programa es terminado).

El módulo 120 de acceso al programa gestiona permitir el acceso a la memoria de un programa que ha sido suspendido o de otro modo hecho terminable. Este acceso permitido puede incluir la reanudación de la ejecución del programa, tal como el lanzamiento de nuevo de un programa que fue terminado o reanudar la ejecución de un programa suspendido. Este permiso de acceso puede también incluir adquirir de nuevo al menos parte de la carga del compromiso de memoria previamente asignada a un programa de modo que otro programa pueda acceder a la memoria, como se discute más adelante con más detalle.

Las Figuras 2A y 2B ilustran unos diagramas de estado de ejemplo 200 que muestran diferentes estados en los que un programa puede estar de acuerdo con una o más realizaciones. El diagrama de estado 200 incluye un estado activo 202, un estado terminable 204, y un estado terminado 206. Un programa puede pasar entre los diferentes estados 202, 204 y 206 como está ilustrado por las flechas. A partir del estado terminable 204 el programa puede pasar al estado activo 202 o al estado terminable 206. A partir del estado terminable 206 el programa puede pasar al estado terminable 204 o al estado activo 202.

En una o más realizaciones como está ilustrado en la Figura 2A, el estado activo 202 y el estado terminable 204 no se solapan. De este modo, en tales realizaciones el programa puede ser activo o terminable, pero no activo y terminable a la vez. Alternativamente, como se ha ilustrado en la Figura 2B, el estado activo 202 y el estado terminable 204 se solapan parcialmente. Por lo tanto, en tales realizaciones el programa puede ser activo pero no terminable, terminable pero no activo, o activo y terminable a la vez.

En el estado activo 202 el programa está funcionando (por ejemplo, está siendo programado para ejecución por uno o más procesadores del sistema o dispositivo que hace funcionar el programa). Mientras tanto, en el estado activo 202 el programa es capaz de acceder a la memoria que ha sido asignada al programa. Desde el estado activo 202 el programa puede pasar al estado terminable 204, tal como cuando el programa es suspendido o de otro modo ser puesto en un estado en el que el programa pueda ser terminado.

En el estado terminable 204 el programa está en un estado en el que el programa puede ser terminado (por ejemplo, el programa es suspendido). Mientras que se pasa del estado activo 202 al estado terminable 204 el programa guarda el estado de procesamiento del programa de modo que si el programa es terminado, el programa puede a continuación ser lanzado de nuevo y devuelto al mismo estado de procesamiento como lo había hecho antes del paso al estado terminable 204. Mientras que en el estado terminable 204 el programa no es capaz de acceder a la memoria que ha sido asignada al programa, y al menos parte de la carga de compromiso de memoria que ha sido asignado al programa puede ser liberada, como se discute más adelante con más detalle.

A partir del estado terminable 204 el programa puede pasar al estado activo 202. El paso al estado activo 202 puede ocurrir debido a diversos sucesos diferentes, tal como una solicitud de un usuario de ejecutar el programa, una solicitud por otro programa de adjuntarse al espacio de memoria del programa, y así sucesivamente. También desde el estado terminable 204 el programa puede pasar al estado terminado 206. El paso al estado terminado 206 puede ocurrir debido a diversos sucesos diferentes, tales como una solicitud de un usuario de cerrar un programa, una determinación que la memoria previamente asignó al programa debe ser usada para uno o más programas adicionales, y así sucesivamente.

En el estado terminado 206, el programa no está ejecutándose. El programa no está cargado en la memoria (por ejemplo, RAM) del sistema o del dispositivo de cálculo que ejecutaría el programa, aunque está almacenado en otro dispositivo de almacenamiento y puede ser cargado en la memoria del sistema o dispositivo de cálculo. A partir del estado terminado 206 el programa puede pasar al estado activo 202. El paso al estado activo 202 puede ocurrir debido a diversos sucesos diferentes tales como una solicitud de un usuario de ejecutar el programa. También desde el estado terminado 204 el programa puede pasar al estado terminable 204. El paso al estado terminable 204 puede ocurrir debido a diversos sucesos diferentes, tal como una solicitud del usuario para cargar pero no ejecutar el programa.

Volviendo a la Figura 1, el controlador 110 de la memoria gestiona la compresión de la memoria usada por un programa. La memoria usada por un programa se refiere a las páginas de memoria que han sido instanciadas por el programa o de otro modo han sido accedidas (por ejemplo, para escribir o almacenar datos) por el programa. La compresión de la memoria de o usada por un programa se refiere a la compresión del contenido de (por ejemplo, datos almacenados en) una o más páginas de memoria asignada al programa. Cualquiera de una variedad de diferentes técnicas de compresión públicas o privadas puede ser usada para comprimir una página de memoria. Por ejemplo, una página de memoria puede ser comprimida usando unos algoritmos de compresión de codificación de longitud de ejecución, algoritmos de compresión LZW, algoritmos de compresión Huffman, y así sucesivamente. Se pueden emplear opcionalmente muchos algoritmos de compresión diferentes, tal como algoritmos de compresión para diferentes tipos de contenidos. Por ejemplo, un algoritmo de compresión puede ser usado para un contenido alfanumérico y/o de texto, otro algoritmo de compresión puede ser usado para un contenido de imágenes, y otro algoritmo de compresión puede ser usado para un contenido de audio. La técnica de compresión usada para comprimir una página de memoria es típicamente un algoritmo de compresión sin pérdidas, aunque en ciertos casos (por ejemplo, para ciertos contenidos de imágenes y/o de contenidos de audio) alternativamente se puede usar un algoritmo de compresión con pérdidas. El particular algoritmo de compresión puede también ser dependiente del dispositivo, en donde un dispositivo más rápido (por ejemplo, con un procesador más potente o más procesadores) puede ser configurado con un algoritmo de compresión más intensivo en procesadores que comprima mejor en tanto que un dispositivo más lento (por ejemplo, con uno o más procesadores menos potentes) puede ser configurado con un algoritmo menos intensivo en procesadores que no comprime tan bien.

La memoria usada por un programa puede ser comprimida en momentos diferentes. En una o más realizaciones la memoria usada por un programa es comprimida mientras que el programa está ejecutándose. El controlador 110 de la memoria puede usar cualquiera de una variedad de reglas o criterios diferentes para determinar qué páginas de memoria son comprimidas. Por ejemplo, todas las páginas de memoria usadas por el programa pueden ser comprimidas, páginas de memoria que no han sido accedidas por el programa para una cantidad umbral de tiempo pueden ser comprimidas, páginas de memoria que se están escribiendo en un archivo de páginas pueden ser comprimidas, y así sucesivamente.

Adicional o alternativamente para comprimir la memoria mientras el programa se está ejecutando, la memoria del programa puede ser comprimida en respuesta al paso del programa a un estado terminable. El controlador 110 de la

memoria puede comprimir todas las páginas de memoria que son usadas por el programa (y todavía no han sido comprimidas), o alternativamente aplicar otras reglas o criterios para determinar qué páginas de memoria son comprimidas. En una o más realizaciones el controlador 110 de la memoria comprime todas las páginas de memoria en la memoria asignada al programa que han sido accedidas (están instanciadas por) el programa no almacenarían ningún contenido para el programa, y por lo tanto no habría un contenido que comprimir.

El módulo 118 de preparación de la terminación del programa gestiona poner el programa en un estado en el que el programa pueda ser terminado. Esto incluye comunicar con el controlador 110 de la memoria para comprimir las páginas de memoria que todavía no han sido comprimidas. Adicionalmente, el módulo 118 de preparación de la terminación del programa determina una cantidad de compromiso de memoria para el programa que puede ser liberada. El módulo 118 de preparación de la terminación del programa almacena las páginas de memoria comprimida en la memoria paginada 108 y/o el dispositivo de almacenamiento 106. La cantidad de memoria comprometida para el programa que puede ser liberada puede variar basado en la manera en la que las páginas de memoria comprimida son almacenadas.

En una o más realizaciones el módulo 118 de preparación de la terminación del programa escribe las páginas de memoria comprimida en uno o más archivos adicionales (distintos del archivo 116 de páginas) del dispositivo 106 de almacenamiento. En esta situación las páginas de memoria comprimida no son almacenadas en la memoria paginada 108 ni en el archivo 116 de páginas, y de este modo todo el compromiso de memoria para el programa puede ser liberado. Por ejemplo, supóngase que 100 Megabits (MB) de memoria son comprometidos al programa, y que las páginas de memoria usadas por el programa pueden ser comprimidas de modo que tomen, en forma comprimida, 40 MB de la memoria. En este ejemplo los contenidos de los 40 MB de la memoria pueden ser escritos en un archivo del dispositivo de almacenamiento, y los 100 MB del compromiso de memoria para el programa pueden ser liberados.

Alternativamente, el módulo 118 de preparación de la terminación del programa escribe las páginas de memoria comprimida en la memoria paginada 108 (después de lo cual una o más páginas pueden ser transferidas al archivo 116 de páginas por el controlador 110 de la memoria). En esta situación las páginas de memoria comprimida son almacenadas en la memoria paginada 108 o en el archivo 116 de páginas, y la cantidad de compromiso de memoria que puede ser liberada es la cantidad comprometida al programa menos la cantidad de memoria en la cual las páginas de memoria fueron comprimidas. Por ejemplo, supóngase que 100 MB de memoria son comprometidos al programa, y que las páginas de memoria usadas por el programa pueden ser comprimidas de modo que tomen, en forma comprimida, 40 MB de la memoria. En este ejemplo, el compromiso de memoria para los 40 MB de memoria comprimida no pueden ser liberados, pero una cantidad igual a la diferencia entre la cantidad de memoria comprometida al programa (100 MB) y la cantidad de memoria en la cual las páginas de memoria fueron comprimidas (40 MB) puede ser liberada (60 MB pueden ser liberados). Este ejemplo supone que las páginas de memoria en las que los datos están comprimidos ellos mismos no toman un compromiso de memoria adicional. Si las páginas de memoria en las que los datos están comprimidos ellos mismos toman un compromiso de memoria adicional, entonces todos los 100 MB de compromiso de memoria para el programa pueden ser liberados.

El módulo 118 de preparación de la terminación del programa notifica al asignador 112 de la memoria de la cantidad de compromiso de memoria para el programa que puede ser liberada, y el compromiso de memoria es liberado. El asignador 112 de la memoria mantiene un registro (por ejemplo, en el dispositivo 106 de almacenamiento, la memoria paginada 108, u otra memoria o almacenamiento del sistema 100) de la cantidad de compromiso de memoria para el programa que es liberada. Este registro de la cantidad de compromiso de memoria para el programa puede ser usado en situaciones en las que la aplicación es posteriormente reanudada, como se discute más adelante con más detalle.

Alternativamente, como se ha reivindicado, en algunas situaciones las páginas de memoria son liberadas en el tiempo de ejecución como se ha discutido antes. En tales situaciones el programa es activo y el compromiso de memoria es liberado en una base por página a medida que las páginas de memoria son comprimidas, y se realiza un intento de adquirir de nuevo el compromiso de memoria en respuesta al programa que accede a una página de memoria comprimida, como se ha discutido antes. En tales situaciones el asignador 112 de la memoria libera unas páginas de memoria a medida que son comprimidas (por ejemplo, y escritas en las páginas de memoria en un almacén comprimido), y además mantiene un registro que identifica qué páginas de memoria comprometidas a un programa han tenido su compromiso de memoria liberado. Este registro (por ejemplo, un registro por página de cada página de memoria) es usado por el asignador 112 de la memoria para determinar si el compromiso de memoria para una página de memoria debe ser adquirido de nuevo si después se accede a la página de memoria. Si el compromiso de memoria no ha sido liberado para una página particular, entonces el compromiso de memoria para esa página particular no necesita ser adquirido de nuevo. No obstante, si el compromiso de la memoria ha sido liberado para una página particular, entonces se realiza un intento de adquirir de nuevo el compromiso de la memoria para esa página particular.

La liberación del compromiso de memoria para el programa da como resultado la conservación del compromiso de memoria en el sistema 100. El compromiso de memoria que fue previamente comprometido a un programa que no es capaz de acceder a la memoria en el momento actual en cualquier modo (por ejemplo, debido a la suspensión del programa) es liberado, permitiendo que el compromiso de memoria sea consumido por otros programas. Por

ejemplo, si 60 MB del compromiso de memoria son liberados debido a la suspensión del programa, esos 60 MB pueden ser asignados a otro programa (y de este modo se convierten en el compromiso de memoria de ese otro programa). Se debería tener en cuenta que si el programa no es suspendido, entonces el programa puede intentar acceder a una página de memoria liberada de compromiso, en cuyo punto el gestor de la memoria intentará adquirir de nuevo el compromiso para la página de memoria, asignar una página física (si el intento de adquirir de nuevo el compromiso tiene éxito), y descomprimir los datos comprimidos en esa página física.

El módulo 120 de acceso al programa gestiona permitir un acceso a la memoria de un programa que ha sido suspendido o terminado. Mientras que en el estado terminable la memoria asignada al programa no puede ser accedida directamente por el programa propiamente dicho u otros programas debido a que el contenido está comprimido y la memoria en la que el contenido (si está descomprimido) sería colocado ha sido liberada. Varios sucesos pueden ocurrir que incluyen solicitudes para acceder al espacio de direcciones de un programa en el estado terminable. En respuesta a tal situación el módulo 120 de acceso al programa gestiona permitir acceso a al menos parte de la memoria previamente asignada al programa intentando obtener de nuevo (también denominado adquirir de nuevo) el compromiso de memoria para al menos parte de la memoria que fue previamente comprometida al programa y liberada. Si el intento de obtener de nuevo el compromiso de memoria para al menos parte de la memoria tiene éxito, entonces el acceso solicitado al espacio de direcciones del programa está permitido. No obstante, si el intento de obtener de nuevo el compromiso de memoria para al menos parte de la memoria no tiene éxito, entonces el programa es terminado (cesa la ejecución del programa). El programa puede opcionalmente ser lanzado de nuevo, lo que da como resultado hacer una nueva asignación de memoria al programa (lo que puede dar como resultado que uno o más de otros programas son pasados a un estado terminable o terminado).

En situaciones en las que la ejecución de un programa ha sido suspendida (o de otro modo situada en un estado no activo), el módulo 120 de acceso al programa puede reanudar la ejecución del programa en respuesta a diversos acontecimientos. En una o más realizaciones estos acontecimientos incluyen una solicitud para reanudar la ejecución del programa, tal como una entrada de usuario que solicita hacer la petición una petición activa (una petición que está actualmente ejecutándose en el dispositivo de cálculo). Adicional o alternativamente estos acontecimientos pueden incluir otras solicitudes, tal como una solicitud de otro programa o código para adjuntar al espacio de direcciones del programa. Estos otros programas o códigos son típicamente ejecutados en un espacio más privilegiado (por ejemplo, en un modo nuclear) que el programa a cuyo espacio de direcciones se está accediendo. Estos otros programas pueden realizar diversas operaciones en el espacio de direcciones, tal como analizar el espacio de direcciones para identificar programas malignos.

En tales situaciones el módulo 120 de acceso al programa puede intentar obtener de nuevo el compromiso de memoria para toda la memoria que fue previamente comprometida al programa y liberada, o alternativamente solamente parte de la memoria que fue previamente comprometida al programa y liberada. Por ejemplo, el módulo 120 de control del programa puede intentar obtener de nuevo el compromiso de memoria para una memoria suficiente para permitir que se acceda a un número particular de páginas de memoria (por ejemplo, las páginas de memoria que incluyen las porciones del espacio de direcciones al que se está accediendo). En tales situaciones el controlador 110 de la memoria asegura que no se puede acceder a las páginas que no han tenido su carga de compromiso obtenida de nuevo (con el fin de impedir la violación de las reglas de los límites de compromiso).

El módulo 120 de acceso al programa puede también determinar si intentar obtener de nuevo parte o todo el compromiso de memoria que fue previamente comprometido al programa y liberado basado en el tipo de acceso que se está realizando. Por ejemplo, si el tipo de acceso es una solicitud del usuario para reanudar la ejecución del programa, entonces el módulo 120 de acceso al programa intenta obtener de nuevo todo el compromiso de memoria que fue previamente comprometido al programa y liberado. Por medio de otro ejemplo, si el tipo de acceso es una solicitud por otro programa para adjuntar a un espacio de direcciones del programa, entonces el módulo 120 de acceso al programa intenta obtener de nuevo un compromiso de memoria suficiente para permitir la descompresión y acceder a las páginas de memoria que incluyen las porciones del espacio de direcciones al que se está accediendo.

El módulo 120 de acceso al programa también puede determinar que, para ciertos tipos de accesos, el compromiso de memoria que fue previamente comprometido al programa y liberado no necesita ser obtenido de nuevo. En una o más realizaciones, si el tipo de solicitud es una solicitud por otro programa para adjuntar a y leer desde un espacio de direcciones del programa, entonces el compromiso de memoria que fue previamente comprometido al programa y liberado no necesita ser obtenido de nuevo. El compromiso de memoria para una página de memoria que ya está respaldada (por ejemplo, en forma comprimida) en el archivo de páginas 116 no necesita ser obtenido de nuevo si la página de memoria debe ser leída debido a que no se harán cambios en la página de memoria que resultaría que daría lugar a la necesidad de guardar la página de memoria en el archivo de páginas o mantenida en la memoria paginada 108. No obstante, si el tipo de solicitud es una solicitud por otro programa para adjuntar a y realizar un acceso sin lectura (por ejemplo, un acceso de escritura, un acceso de bloqueo de página de memoria, etc), entonces el módulo 120 de acceso al programa intenta obtener de nuevo un suficiente compromiso de memoria para permitir la descompresión y el acceso a las páginas de memoria que incluyen las porciones del espacio de direcciones al que se está accediendo.

En otras situaciones como se ha discutido antes, en donde el gestor de la memoria sigue un enfoque de liberación en tiempo de ejecución, el programa permanece activo y el módulo 120 de acceso al programa no necesita reanudar la ejecución del programa. Más bien, se recibe cuando una solicitud (por ejemplo, por el programa, o una solicitud por otro programa para adjuntar a y leer desde un espacio de direcciones del programa) para acceder a una página de memoria del programa liberada de compromiso, el compromiso es vuelto a cargar, una página física es asignada, y los datos son descomprimidos desde el almacén comprimido en la página física. Si la carga de compromiso falla (la carga de compromiso no puede ser adquirido de nuevo), entonces surge una excepción (y/o el programa es terminado) como se ha discutido antes.

La Figura 3 es un diagrama de flujos que ilustra un proceso de ejemplo 300 para conservar el compromiso de memoria cuando un programa es terminable de acuerdo con una o más realizaciones. El proceso 300 es realizado por un gestor de la memoria y el módulo de preparación de terminación del programa, tal como el gestor 104 de la memoria y el módulo 118 de preparación de la terminación del programa de la Figura 1, y puede ser aplicado en soporte lógico, microprogramación cableada, soporte físico, o una combinación de ellos. El proceso 300 se muestra como un conjunto de acciones y no está limitado al orden mostrado para realizar las operaciones de las diversas acciones. El proceso 300 es un proceso de ejemplo para conservar el compromiso de memoria cuando un programa es terminable; discusiones adicionales de conservación del compromiso de memoria cuando un programa es terminable están aquí incluidas con referencia a diferentes figuras.

En el proceso 300 las páginas de memoria de un programa son comprimidas mientras que el programa está activo (acción 302). Al menos algunas de las páginas de memoria de un programa pueden ser comprimidas por el controlador de la memoria para conservar la memoria mientras que el programa está ejecutándose como se ha discutido antes. No obstante, en realizaciones alternativas no reivindicadas, no ocurre tal compresión de páginas de memoria mientras que el programa está activo.

En algún momento se hace una determinación de que un programa está en un estado desde el cual el programa puede ser terminado (acción 304). Como se ha discutido antes, el programa puede estar en un estado desde el cual el programa puede ser terminado siendo pasado a un estado terminable, o estando a la vez en un estado activo y terminable.

Se identifica un subconjunto de las páginas de memoria del programa para ser comprimido (acción 306). El subconjunto de las páginas de memoria para ser comprimido son las páginas de memoria incluidas en la memoria asignada al programa a las que se ha accedido por (son instanciadas por) el programa y que todavía no han sido comprimidas (por ejemplo, en la acción 302).

El subconjunto identificado de las páginas de memoria es comprimido (acción 308). Las páginas de memoria pueden ser comprimidas usando cualquiera de una variedad de diferentes algoritmos o técnicas de compresión como se ha discutido antes.

Una cantidad de compromiso de memoria para el programa que puede ser liberada es determinada (acción 310). La cantidad de compromiso de memoria que puede ser liberado es igual al valor obtenido sustrayendo la cantidad de memoria en la que las páginas de memoria fueron comprimidas de la cantidad del compromiso de memoria para el programa. El compromiso de memoria para las páginas no instanciadas (páginas de memoria del programa que no fueron usadas por el programa) es incluido en la cantidad del compromiso de memoria para el programa que puede ser liberado. La cantidad de compromiso de memoria que puede ser liberado puede ser menor que todo el compromiso de memoria para el programa (por ejemplo, en situaciones en las que la memoria comprimida es almacenada en una memoria paginada o archivo de páginas), o todo el compromiso de memoria para el programa (por ejemplo, en situaciones en las que la memoria comprimida es almacenada en un archivo adicional en un dispositivo de almacenamiento distinto del archivo de páginas, o en situaciones en las que la memoria comprimida ha cargado un compromiso para ella misma).

Se debería tener en cuenta que en algunas situaciones cierto compromiso de memoria para el programa no puede ser liberado debido a las garantías hechas a otros programas o código (por ejemplo, controladores de modo nuclear). Tal compromiso de memoria está excluido de la cantidad de compromiso de memoria que puede ser liberado como se ha determinado en la acción 310. De este modo, la cantidad de compromiso de memoria que puede ser liberado es igual al valor obtenido sustrayendo de la cantidad de compromiso de memoria para el programa, la cantidad de memoria en la que las páginas de memoria fueron comprimidas y la cantidad del compromiso de memoria que no puede ser liberado debido a las garantías hechas a otros programas o códigos.

La cantidad de compromiso de memoria determinada en la acción 310 es liberada (acción 312). La liberación del compromiso de memoria permite que la cantidad del compromiso de memoria sea consumida por otros programas que están ejecutándose en el dispositivo de cálculo. La carga del compromiso de páginas de memoria del programa que no fueron usadas por el programa es reducida a cero, y la carga de compromiso para páginas de memoria que fueron usadas por el programa (y posteriormente comprimidas) es reducida en la relación de compresión.

Se mantiene un registro de la cantidad del compromiso de memoria que es liberado (acción 314). Este registro permite que se realice a continuación una determinación sobre cómo cuánto compromiso de memoria ha de

obtenerse de nuevo para un programa. En una o más realizaciones el registro incluye una indicación de qué páginas de memoria comprometidas a un programa han tenido su compromiso de memoria liberado, permitiendo que el gestor de la memoria conozca si el compromiso de memoria para una página de memoria debe ser adquirido de nuevo si se accede a la página de memoria como se ha discutido antes.

5 La Figura 4 es un diagrama de flujos que ilustra un proceso de ejemplo 400 para permitir el acceso a la memoria de un programa que ha sido suspendido o de otro modo hecho terminable de acuerdo con una o más realizaciones. El proceso 400 es realizado por un gestor de la memoria y un módulo de acceso al programa, tal como el gestor 104 de la memoria y el módulo 120 de acceso al programa de la Figura 1, y puede ser aplicado en soporte lógico, microprogramación cableada, soporte físico, o una combinación de ellos. El proceso 400 se muestra como un conjunto de acciones y no está limitado al orden mostrado para realizar las operaciones de las diversas acciones. El proceso 400 es un proceso de ejemplo para permitir el acceso a la memoria de un programa que ha sido suspendido o de otro modo hecho terminable; discusiones adicionales de permitir el acceso a la memoria de un programa que ha sido suspendido o de otro modo hecho terminable están incluidas aquí con referencia a diferentes figuras.

15 En el proceso 400 se hace una determinación de que se va a acceder a una o más páginas de memoria de un programa que ha tenido al menos parte de su compromiso de memoria liberado (acción 402). Se puede acceder a una o más páginas de memoria como parte de diversos sucesos como se ha discutido antes.

20 Se hace un intento para obtener de nuevo el compromiso de memoria para al menos una o más páginas de memoria del programa (acción 404). Este intento puede ser realizado para obtener de nuevo todo el compromiso de memoria que fue liberado (por ejemplo, como está indicado en el registro mantenido en la acción 314 de la Figura 3), o un subconjunto del compromiso de memoria (por ejemplo, compromiso de memoria para páginas de memoria particulares a las que se está accediendo) como se ha discutido antes.

25 El proceso 400 sigue su curso basado en si el intento en la acción 404 tiene éxito (acción 406). Si el intento en la acción 404 tiene éxito, entonces se permite el acceso a una o más páginas de memoria (acción 408). No obstante, si el intento no tiene éxito, entonces el programa es terminado (acción 410) y opcionalmente es lanzado de nuevo (acción 412). Se debería tener en cuenta que terminando y lanzando de nuevo el programa la memoria es asignada al programa en respuesta a una solicitud de asignación de memoria para el programa, lo que puede dar como resultado que uno o más de otros programas en el dispositivo de cálculo sean terminados.

30 Las acciones realizadas en la terminación del programa pueden variar basadas en la aplicación. En una o más realizaciones la terminación del programa incluye desmontar el espacio de direcciones del programa (como si el programa estuviera saliendo), y la activación de la terminación de la fibra fuerte para todas las fibras en el proceso que es el programa en ejecución (por ejemplo, que pone en cola mensajes a las fibras para sacarlas de su estado suspendido y terminar).

35 En una o más realizaciones se tiene cuidado al terminar un programa de modo que se mantenga la sincronización con otros programas en el dispositivo de cálculo. El programa es terminado en la acción 410 de una manera que es congruente con lo que el programa así como cualquier otro programa (por ejemplo que puede estar adjunto al programa que está siendo terminado) espera. Por ejemplo, el programa es terminado en la acción 410 solamente si ningún otro programa está actualmente adjunto al espacio de direcciones del programa.

40 Volviendo a la Figura 1, en una o más realizaciones el sistema 102 de gestión de la memoria proporciona un soporte adicional para facilitar guardar el compromiso de memoria. El sistema 102 de gestión de la memoria permite que un programa 114 indique que una particular cantidad de compromiso de memoria previamente asignado al programa 114 pueda ser liberado si el sistema 102 de gestión de la memoria lo desea. Proporcionando tal indicación el programa 114 está reconociendo que el programa 114 puede aceptar una pérdida de compromiso de memoria previamente asignado al programa 114. Tal indicación puede ser proporcionada, por ejemplo, invocando una interfaz de programación de aplicación (API) expuesta por el asignador 112 de la memoria. La provisión de la indicación permite al asignador 112 de la memoria liberar parte del compromiso de memoria previamente asignado al programa 45 114 y asignar el compromiso de memoria a otro programa que solicite asignación de memoria.

50 Las técnicas aquí discutidas soportan diversas situaciones de uso. Las técnicas aquí discutidas conservan un compromiso de memoria en un dispositivo de cálculo reduciendo la cantidad del compromiso de memoria asignado a programas en el dispositivo de cálculo. Esto aumenta el número de programas que pueden ser ejecutados o suspendidos en un dispositivo de cálculo en cualquier momento dado sin tener que terminar los programas. Esto también disminuye la cantidad de memoria paginada y/o el espacio del dispositivo de almacenamiento (para el archivo de páginas) necesario para permitir que un número dado de programas sean ejecutados o suspendidos en un dispositivo de cálculo en cualquier momento dado sin tener que terminar cualquiera de los programas.

55 Aunque aquí se discute la funcionalidad particular con referencia a módulos particulares, se debería tener en cuenta que la funcionalidad de los módulos individuales aquí discutidos puede ser separada en múltiples módulos, y/o al menos alguna funcionalidad de los múltiples módulos puede ser combinada en un único módulo. Adicionalmente, un módulo particular aquí discutido realizando una acción incluye ese particular módulo propiamente dicho realizando la acción, o alternativamente ese módulo particular invocando o de otro modo accediendo a otro componente o módulo

que realiza la acción (o realiza la acción en conjunción con ese módulo particular). De este modo, un módulo particular que realiza una acción incluye el módulo particular propiamente dicho realizando la acción y/o otro módulo invocado o de otro modo accedido por ese módulo particular que realiza la acción.

5 La Figura 5 ilustra un sistema de ejemplo generalmente en 500 que incluye un dispositivo de cálculo 502 de ejemplo que es representativo de uno o más sistemas y/o dispositivos que pueden aplicar las diversas técnicas aquí descritas. El dispositivo de cálculo 502 puede ser, por ejemplo, un servidor de un proveedor del servicio, un dispositivo asociado con un cliente (un dispositivo cliente), y un sistema en chip, y/o cualquier otro dispositivo de cálculo o sistema de cálculo adecuados.

10 El dispositivo de cálculo 502 de ejemplo como está ilustrado incluye un sistema de procesamiento 504, uno o más medios 506 legibles por ordenador, y una o más interfaces I/O 508 que están acoplados comunicativamente uno con otro. Aunque no mostrado, el dispositivo de cálculo 502 puede además incluir un sistema de barra colectora u otros datos y un sistema de transferencia de órdenes que acopla los diversos componentes, uno con otro. Un sistema de barra colectora puede incluir cualquiera o una combinación de diferentes estructuras de barra colectora, tal como una barra colectora de memoria o controlador de memoria, una barra colectora periférica, una barra colectora en serie universal, y/o un procesador o barra colectora local que utiliza cualquiera de una variedad de arquitecturas de barra colectora. También se contemplan una variedad de otros ejemplos tales como líneas de control y datos.

15 El sistema de procesamiento 504 es representativo de la funcionalidad para realizar una o más operaciones usando un soporte físico. En consecuencia, el sistema de procesamiento 504 está ilustrado como incluyendo elementos de soporte físico 510 que pueden ser configurados como procesadores, bloques funcionales, y así sucesivamente. Esto puede incluir la aplicación en soporte físico como un circuito integrado de aplicación específica u otro dispositivo lógico formado que usa uno o más semiconductores. Los elementos de soporte físico 510 no están limitados por los materiales de los que están formados o los mecanismos de procesamiento ahí empleados. Por ejemplo, los procesadores pueden estar compuestos por un o unos semiconductores y/o transistores (por ejemplo, circuitos electrónicos integrados (ICs)). En tal contexto, las instrucciones ejecutables por un procesador pueden ser unas instrucciones electrónicamente ejecutables.

20 Los medios 506 legibles por ordenador están ilustrados como incluyendo una memoria/almacenamiento 512. La memoria/almacenamiento 512 representa una capacidad de memoria/almacenamiento asociada con uno o más medios legibles por ordenador. La memoria/almacenamiento 512 puede incluir unos medios volátiles (tal como una memoria de acceso aleatorio (RAM)) y/o unos medios no volátiles (tal como una memoria de sólo lectura (ROM), una memoria Flash, discos ópticos, discos magnéticos, y así sucesivamente). La memoria/almacenamiento 512 puede incluir unos medios fijos (por ejemplo, RAM, ROM, un disco duro fijo, y así) y también medios extraíbles (por ejemplo, una memoria Flash, un disco duro extraíble, un disco óptico, y así sucesivamente). Los medios 506 legibles por ordenador pueden estar configurados en una variedad de otras formas como se describirá más adelante.

30 La o las interfaces entrada/salida 508 son representativas de la funcionalidad para permitir a un usuario introducir instrucciones e información al dispositivo de cálculo 502, y también permitir que la información sea presentada al usuario y/o otros componentes o dispositivos usando diversos dispositivos de entrada/salida. Los ejemplos de los dispositivos de entrada incluyen un teclado, un dispositivo de control del cursor (por ejemplo, un ratón), un micrófono (por ejemplo, para entradas de voz), un dispositivo de exploración, una funcionalidad táctil (por ejemplo, capacitiva u otros sensores que están configurados para detectar el toque físico), una cámara (por ejemplo, que puede emplear longitudes de onda visibles o no visibles tales como frecuencias infrarrojas para detectar el movimiento que no implica el tacto como gestos), y así sucesivamente. Ejemplos de dispositivos de salida incluyen un dispositivo de visualización (por ejemplo, un monitor o proyector), altavoces, una impresora, una tarjeta de red, un dispositivo de respuesta táctil, y así sucesivamente. De este modo, el dispositivo de cálculo 502 puede ser configurado en una variedad de modos como se describirá más adelante para soportar la interacción con el usuario.

35 El dispositivo de cálculo 502 incluye también un sistema 514 de gestión de la memoria. El sistema 514 de gestión de la memoria proporciona una funcionalidad diversa para conservar la memoria reduciendo la carga de compromiso de memoria como se ha discutido antes. El sistema 514 de gestión de la memoria puede aplicar, por ejemplo, el sistema 102 de gestión de la memoria de la Figura 1. Aunque está ilustrado como separado de los medios 506 legibles por ordenador, se debería tener en cuenta que el sistema 514 de gestión de la memoria puede alternativamente ser aplicado como parte de los medios 506 legibles por ordenador.

40 Diversas técnicas pueden ser descritas aquí en el contexto general del soporte lógico, elementos del soporte físico, o módulos de programa. Generalmente, tales módulos incluyen rutinas, programas, objetos, elementos, componentes, estructuras de datos, y así sucesivamente que realizan unas tareas particulares o aplican tipos de datos abstractos particulares. Los términos "módulo", "funcionalidad", y "componente" como se usan aquí generalmente representan soporte lógico, microprogramación cableada, soporte físico, o una combinación de ellos. Las características de las técnicas aquí descritas son independientes de la plataforma, lo que significa que las técnicas pueden ser aplicadas en una variedad de plataformas de cálculo que tengan una variedad de procesadores.

Una aplicación de los módulos y técnicas descritos pueden ser almacenados en y transmitidos a través de alguna forma de medios legibles por ordenador. Los medios legibles por ordenador pueden incluir una variedad de medios

que pueden ser accedidos por el dispositivo de cálculo 502. A modo de ejemplo, y no de limitación, los medios legibles por ordenador pueden incluir “medios de almacenamiento legibles por ordenador” y “medios de señales legibles por ordenador”.

5 Los “medios de almacenamiento legibles por ordenador” se refieren a medios y/o dispositivos que permiten el almacenamiento permanente de información y/o almacenamiento que es tangible, en contraste con una simple transmisión de señales, ondas portadoras, o señales en sí mismas. De este modo, los medios de almacenamiento legibles por ordenador se refieren a medios que no llevan señales. Los medios de almacenamiento legibles por ordenador incluyen un soporte físico tal como volátil y no volátil, unos medios extraíbles y no extraíbles y/o unos dispositivos de almacenamiento aplicados en un método o tecnología apropiada para almacenamiento de información tal como instrucciones legibles por ordenador, estructuras de datos, módulos de programa, elementos lógicos / circuitos, u otros datos. Ejemplos de medios de almacenamiento legibles por ordenador pueden incluir, pero no están limitados a, RAM, ROM, EEPROM, memoria Flash u otra tecnología de memoria, CD-ROM, discos digitales versátiles (DVD) u otro almacenamiento óptico, discos duros, cassetes magnéticas, cintas magnéticas, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, u otros dispositivos de almacenamiento, medios tangibles, o artículos de fabricación apropiada para almacenar la información deseada y que puedan ser accedidos por un ordenador.

20 Los “medios de señales legibles por ordenador” se refieren a un medio portador de señales que está configurado para transmitir instrucciones al soporte físico del dispositivo de cálculo 502, tal como por medio de una red. Los medios de señales típicamente pueden incorporar instrucciones legibles por ordenador, estructuras de datos, módulos de programas, u otros datos en una señal de datos modulados tal como ondas portadoras, señales de datos, u otro mecanismo de transporte. Los medios de señales también incluyen cualquier medio de entrega de información. El término “señal de datos modulados” significa una señal que tiene una o más de sus características fijadas o cambiadas de tal manera como para codificar información en la señal. A modo de ejemplo, y no de limitación, los medios de comunicación incluyen medios por cable tal como una red por cable o una conexión directa por cable, y unos medios inalámbricos tales como acústicos, RF, infrarrojos, y otros medios inalámbricos.

30 Como previamente se ha descrito, los elementos 510 de soporte físico y los medios 506 legibles por ordenador son representativos de instrucciones, módulos, dispositivos lógicos programables y/o lógica del dispositivo fijo aplicado en una forma de soporte físico que puede ser empleado en algunas realizaciones para aplicar al menos algunos aspectos de las técnicas aquí descritas. Los elementos de soporte físico pueden incluir componentes de un circuito integrado o un sistema en chip, un circuito integrado de aplicación específica (ASIC), una matriz de puerta programable de campo (FPGA), un dispositivo lógico programable complejo (CPLD), y otras aplicaciones en silicio u otros dispositivos de soporte físico. En este contexto, un elemento de soporte físico puede operar como un dispositivo de procesamiento que realiza tareas de programas definidas por instrucciones, módulos, y/o una lógica incorporada por el elemento de soporte físico así como un dispositivo de soporte físico utilizado para almacenar instrucciones para ejecución, por ejemplo, los medios de almacenamiento legibles por ordenador previamente descritos.

40 Unas combinaciones de lo anterior pueden también ser empleadas para aplicar las diversas técnicas y módulos aquí descritos. Por lo tanto, el soporte lógico, el soporte físico, o los módulos de programas y otros módulos de programas pueden ser aplicados como una o más instrucciones y/o lógica incorporada en alguna forma de medios de almacenamiento legibles por ordenador y/o por uno o más elementos 510 de soporte físico. El dispositivo de cálculo 502 puede ser configurado para aplicar instrucciones particulares y/o funciones que corresponden a los módulos de soporte lógico y/o soporte físico. Por consiguiente, la aplicación de módulos como un módulo que es ejecutable por el dispositivo de cálculo 502 como soporte lógico puede ser conseguida, al menos parcialmente, en soporte físico, por ejemplo, a través del uso de medios de almacenamiento legibles por ordenador y/o elementos 510 de soporte físico del sistema de procesamiento. Las instrucciones y/o funciones pueden ser ejecutables/operables por uno o más artículos de fabricación (por ejemplo, uno o más dispositivos de cálculo 502 y/o sistemas de procesamiento 504) para aplicar las técnicas, módulos, y ejemplos aquí descritos.

50 Como además está ilustrado en la Figura 5, el sistema 500 de ejemplo permite unos entornos ubicuos para experiencia de un usuario sin problemas cuando ejecuta aplicaciones en un ordenador personal (PC), un dispositivo de televisión, y/o un dispositivo móvil. Los servicios y aplicaciones se ejecutan sustancialmente igual en los tres entornos para una experiencia de usuario común cuando se pasa de un dispositivo al siguiente mientras que se utiliza una aplicación, jugando a un juego de vídeo, viendo un vídeo, y así.

55 En el sistema 500 de ejemplo, varios dispositivos están interconectados a través de un dispositivo de cálculo central. El dispositivo de cálculo central puede ser local para los varios dispositivos o puede estar situado lejos de los varios dispositivos. En una o más realizaciones el dispositivo de cálculo central puede ser una nube de uno o más ordenadores servidores que están conectados a los varios dispositivos a través de una red, la internet, u otro enlace de comunicación de datos.

60 En una o más realizaciones esta arquitectura de interconexión permite que una funcionalidad sea entregada a través de varios dispositivos para proporcionar una experiencia común y sin problemas a un usuario de los varios dispositivos. Cada uno de los varios dispositivos puede tener diferentes exigencias y capacidades físicas, y el

dispositivo de cálculo central usa una plataforma para permitir la entrega de una experiencia al dispositivo que está adaptado al dispositivo y además común a todos los dispositivos. En una o más realizaciones se crea una clase de dispositivos objetivo y las experiencias se adaptan a la clase genérica de dispositivos. Una clase de dispositivos puede ser definida por características físicas, tipos de uso, u otras características comunes de los dispositivos.

5 En diversas aplicaciones el dispositivo de cálculo 502 puede adoptar una variedad de diferentes configuraciones, tal como para el uso del ordenador 516, el dispositivo móvil 518, y la televisión 520. Cada una de estas configuraciones incluye unos dispositivos que pueden tener unas estructuras y capacidades generalmente diferentes, y así el dispositivo de cálculo 502 puede ser configurado de acuerdo con una o más de las diferentes clases de dispositivos. Por ejemplo, el dispositivo de cálculo 502 puede ser aplicado como la clase de un ordenador 516 de un dispositivo
10 que incluye un ordenador personal, un ordenador multipantalla, un ordenador portátil, un libro de notas, y así sucesivamente.

El dispositivo de cálculo 502 puede también ser aplicado como la clase de dispositivo móvil 518 que incluye dispositivos móviles, tales como un teléfono móvil, un reproductor de música portátil, un dispositivo de juegos portátil, un ordenador de tableta, un ordenador multipantalla, y así sucesivamente. El dispositivo de cálculo 502
15 puede también ser aplicado como la clase de dispositivo de televisión 520 que incluye unos dispositivos que tienen o están conectados a generalmente pantallas mayores en ambientes de visión informales. Estos dispositivos incluyen televisiones, cajas encima de objetos, consolas de juegos, y así sucesivamente.

Las técnicas aquí descritas pueden ser soportadas por estas diversas configuraciones del dispositivo de cálculo 502 y no están limitadas por los ejemplos específicos de las técnicas aquí descritas. Esta funcionalidad puede también
20 ser aplicada toda o en parte mediante el uso de un sistema distribuido tal como una “nube” 522 por medio de una plataforma 524 como se describe más adelante.

La nube 522 incluye y/o es representativa de una plataforma 524 para recursos 526. La plataforma 524 abstrae la funcionalidad subyacente del soporte físico (por ejemplo, servidores) y recursos del soporte lógico de la nube 522. Los recursos 526 pueden incluir aplicaciones y/o datos que pueden ser utilizados mientras el procesamiento
25 informático es ejecutado con servidores que están alejados del dispositivo de cálculo 502. Los recursos 526 pueden también incluir servicios proporcionados en internet y/o a través de una red de suscriptores tal como una red celular o Wi-Fi.

La plataforma 524 puede abstraer recursos y funciones para conectar el dispositivo de cálculo 502 con otros dispositivos de cálculo. La plataforma 524 puede también servir para abstraer la escalada de recursos para
30 proporcionar un correspondiente nivel de escala a la demanda encontrada de recursos 526 que son aplicados por medio de la plataforma 524. En consecuencia, en una realización del dispositivo interconectado la aplicación de la funcionalidad aquí descrita puede ser distribuida en todo el sistema 500. Por ejemplo, la funcionalidad puede ser aplicada en parte en el dispositivo de cálculo 502 así como por medio de la plataforma 524 que abstrae la funcionalidad de la nube 522.

35 En estas discusiones se han descrito diversas realizaciones diferentes. Hay que apreciar y entender que cada realización aquí descrita puede ser usada por sí misma o en conexión con una o más de otras realizaciones aquí descritas. Aspectos adicionales de las técnicas aquí discutidas se refieren a una o más de las siguientes realizaciones.

Aunque la materia objeto ha sido descrita en un lenguaje específico de las características estructurales y/o acciones metodológicas, se ha de entender que la materia objeto definida en las reivindicaciones anejas no está
40 necesariamente limitada a las características o acciones específicas antes descritas. Más bien, las características y acciones específicas antes descritas están descritas como ejemplos de formas de aplicación de las reivindicaciones.

REIVINDICACIONES

1. Un método (300) de reducir una cantidad de compromiso de memoria para un programa en un dispositivo de cálculo (502), comprendiendo el método:
- 5 determinar (304) que el programa está en un estado a partir del cual el programa puede ser terminado, habiendo sido comprometida al programa una cantidad de memoria, siendo la cantidad de memoria un compromiso de memoria que garantiza la disponibilidad de una cantidad de memoria asignada al programa, habiendo sido usado por el programa un subconjunto de la cantidad de memoria comprometida al programa, e incluyendo la cantidad de memoria comprometida al programa varias páginas de memoria;
- 10 comprimir (308) el contenido de las páginas de memoria en el subconjunto de memoria usado por el programa, en donde la compresión comprende comprimir, por un controlador de la memoria del dispositivo de cálculo (502), uno o más del subconjunto las páginas de memoria mientras que el programa se está ejecutando y antes de que el programa sea pasado al estado a partir del cual el programa puede ser terminado;
- determinar (310) una cantidad del compromiso de memoria para el programa que puede ser liberada;
- 15 liberar (312), después de que el subconjunto de memoria usado por el programa ha sido comprimido, la cantidad determinada del compromiso de memoria para el programa, efectiva para reducir la cantidad de memoria garantizada para estar disponible al programa en una cantidad menor de la que es normalmente asignada al programa; y
- mantener un registro del compromiso de memoria liberado.
2. El método (300) citado en la reivindicación 1, el estado a partir del cual el programa puede ser terminado comprendiendo un estado terminable en el que un estado de procesamiento del programa ha sido guardado y al cual el programa puede ser posteriormente recuperado.
3. El método (300) citado en la reivindicación 1, el estado a partir del cual el programa puede ser terminado comprendiendo un estado activo.
- 25 4. El método (300) citado en cualquiera de las reivindicaciones 1 a 3, la compresión del contenido de las páginas de memoria comprendiendo también la compresión, por un controlador de la memoria del dispositivo de cálculo (502) en respuesta a determinar que el programa ha sido pasado al estado a partir del cual el programa puede ser terminado, de una o más del subconjunto de las páginas de memoria.
5. El método (300) citado en cualquiera de las reivindicaciones 1 a 4, el subconjunto de memoria usada por el programa comprendiendo páginas de memoria accedidas por o instanciadas por el programa.
- 30 6. El método (300) citado en cualquiera de las reivindicaciones 1 a 5, el subconjunto de memoria usada por el programa comprendiendo menos que toda la memoria comprometida al programa.
7. El método (300) citado en cualquiera de las reivindicaciones 1 a 6, la cantidad del compromiso de memoria para el programa que puede ser liberada comprendiendo toda la memoria comprometida al programa.
- 35 8. El método (300) citado en cualquiera de las reivindicaciones 1 a 6, la cantidad del compromiso de memoria para el programa que puede ser liberada comprendiendo una cantidad igual a toda la memoria comprometida para el programa menos una cantidad de memoria usada en la cual el contenido comprimido es almacenado.
9. El método (300) citado en cualquiera de las reivindicaciones 1 a 8 comprende además, después de liberar el compromiso de memoria por la cantidad de memoria que puede ser liberada:
- determinar que una o más páginas de memoria del programa deben ser accedidas;
- 40 intentar obtener de nuevo el compromiso de memoria para al menos una o más páginas de memoria del programa usando el registro;
- permitir, en respuesta al intento de obtener de nuevo el compromiso de memoria que tiene éxito, el acceso a una o más páginas de memoria; y
- 45 terminar y lanzar de nuevo el programa en respuesta al intento de obtener de nuevo el compromiso de memoria que no tiene éxito.
10. Un dispositivo de cálculo (502) que incluye un sistema (514) de gestión de la memoria configurado para conservar el compromiso de memoria en el dispositivo de cálculo (502), comprendiendo el sistema de gestión de la memoria:
- un módulo de preparación de la terminación del programa configurado para:

- 5 determinar (304) que un programa está en un estado a partir del cual el programa puede ser terminado, habiendo sido comprometida al programa una cantidad de memoria, siendo la cantidad de memoria un compromiso de memoria que garantiza la disponibilidad de una cantidad de memoria asignada al programa, habiendo sido usado por el programa un subconjunto de la cantidad de memoria comprometida al programa, e incluyendo la cantidad de memoria comprometida al programa varias páginas de memoria;
- tener (308) el contenido de las páginas de memoria en el subconjunto de memoria usado por el programa comprimido, y para tener una o más del subconjunto de las páginas de memoria comprimida mientras que el programa está ejecutándose y antes de que el programa sea pasado al estado a partir del cual el programa puede ser terminado;
- 10 determinar (310) una cantidad del compromiso de memoria para el programa que puede ser liberado;
- liberar, después de que el subconjunto de memoria usado por el programa haya sido comprimido, la cantidad de compromiso de memoria determinada para el programa, efectiva para reducir la cantidad de memoria usada por el programa; y
- mantener un registro del compromiso de memoria liberada.
- 15 11. El dispositivo de cálculo (502) citado en la reivindicación 10, que además comprende un módulo de acceso al programa configurado para, después de la liberación de la carga de compromiso para la cantidad de memoria que puede ser liberada:
- determinar que una o más páginas de memoria del programa van a ser accedidas;
- 20 intentar obtener de nuevo el compromiso de memoria para al menos la una o más páginas de memoria del programa, usando el registro;
- permitir, en respuesta al intento de obtener de nuevo el compromiso de memoria con éxito, la una o más páginas de memoria que pueden ser accedidas, y;
- terminar y lanzar de nuevo el programa en respuesta al intento de obtener de nuevo el compromiso de memoria sin éxito.
- 25 12. El dispositivo de cálculo (502) citado en la reivindicación 11, incluyendo el sistema de gestión de la memoria además un asignador de la memoria configurado para recibir del programa una indicación de que un compromiso de la cantidad de memoria previamente asignada puede ser liberado si se desea por el asignador de la memoria, y liberar la cantidad de compromiso de memoria previamente asignada y asignar el compromiso de la cantidad de memoria previamente asignada a otro programa del dispositivo de cálculo (502).
- 30 13. El dispositivo de cálculo (502) citado en la reivindicación 11 o la reivindicación 12, estando el módulo de preparación de la terminación del programa configurado además para tener comprimido el contenido de las páginas de memoria, por un controlador de la memoria del sistema de gestión de la memoria, también en respuesta a la determinación de que el programa ha sido pasado al estado a partir del cual el programa puede ser terminado.
- 35 14. El dispositivo de cálculo (502) citado en cualquiera de las reivindicaciones 11 a 13, en donde para intentar obtener de nuevo el compromiso de memoria para al menos una o más páginas de memoria se debe intentar obtener de nuevo el compromiso de memoria para cada una de una o más páginas en una base por página a medida que las solicitudes de acceso son recibidas.
- 40 15. El dispositivo de cálculo (502) citado en cualquiera de las reivindicaciones 11 a 14, en donde para determinar que una o más páginas de memoria del programa deben ser accedidas es determinar que una o más páginas de memoria deben ser accedidas adjuntando el código a un espacio de direcciones del programa.

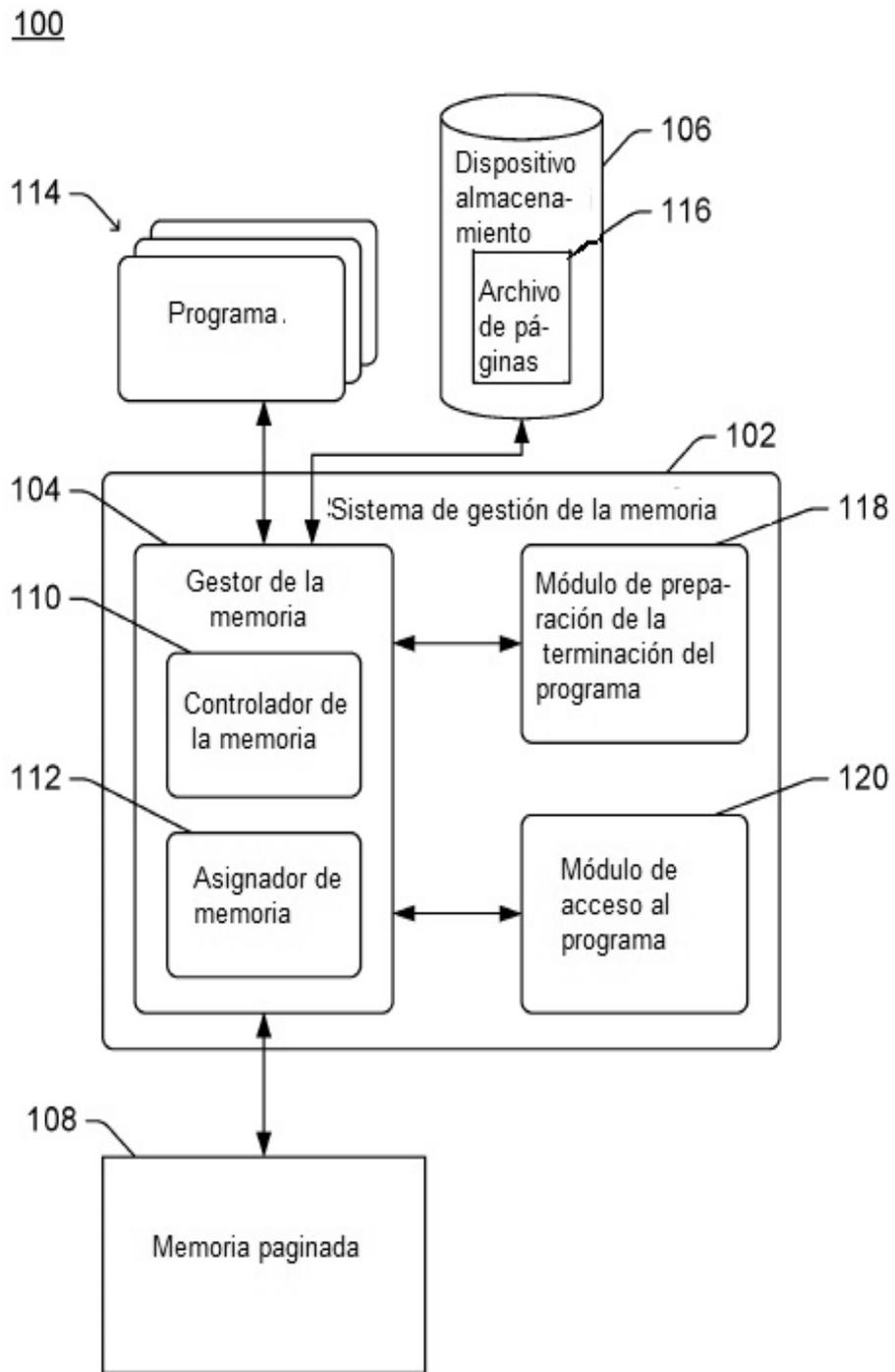


Fig. 1

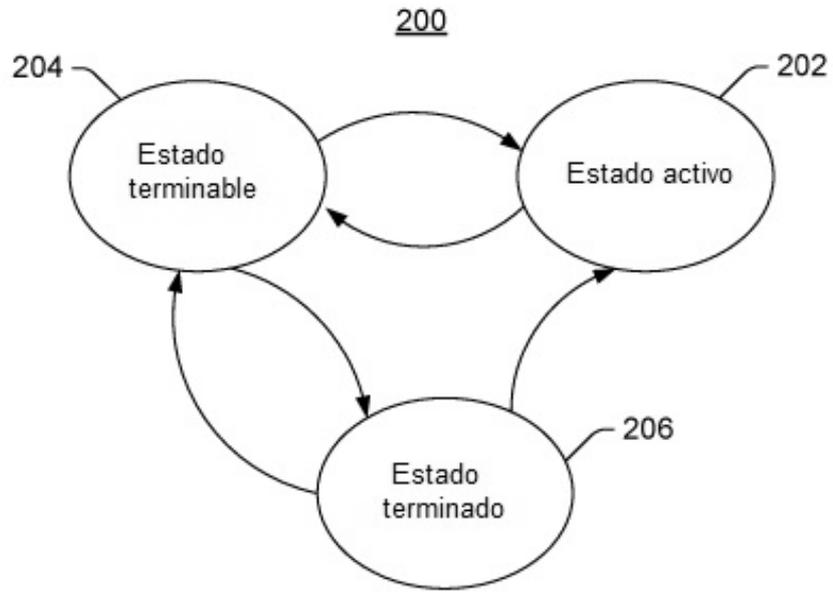


Fig. 2A

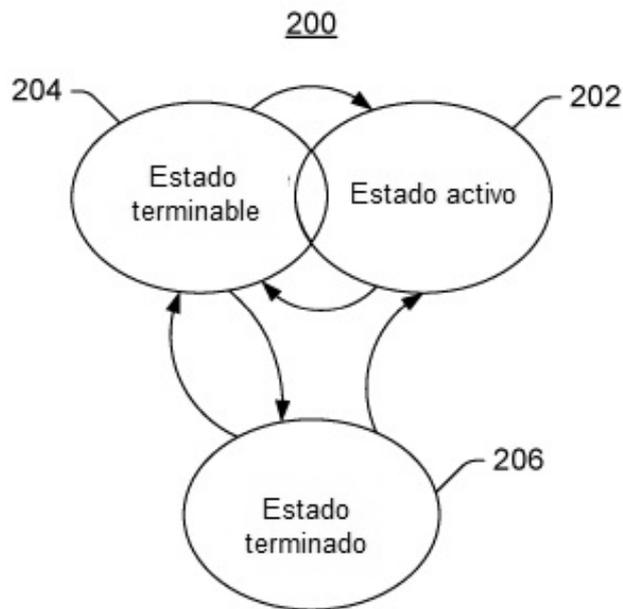


Fig. 2B

300



Fig. 3

400

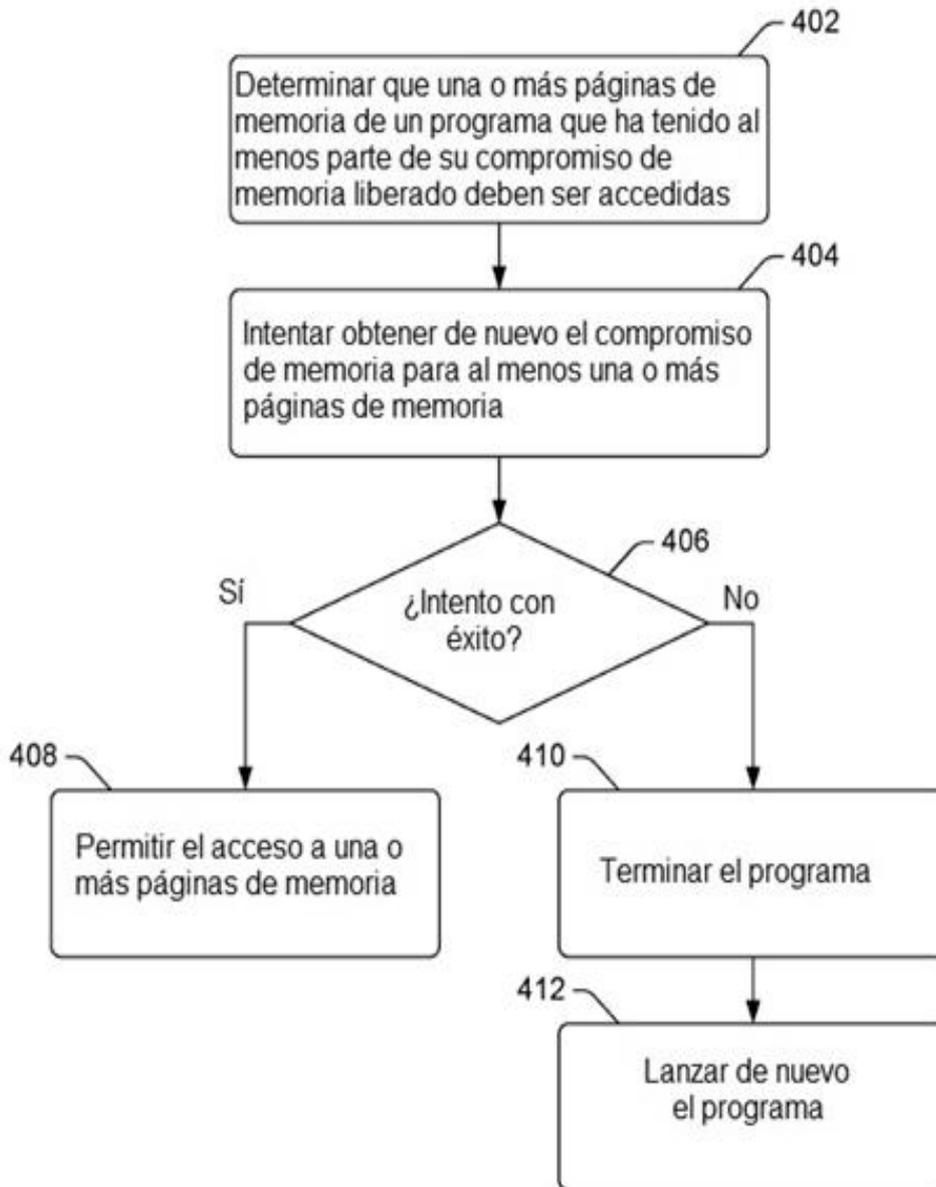


Fig. 4

500

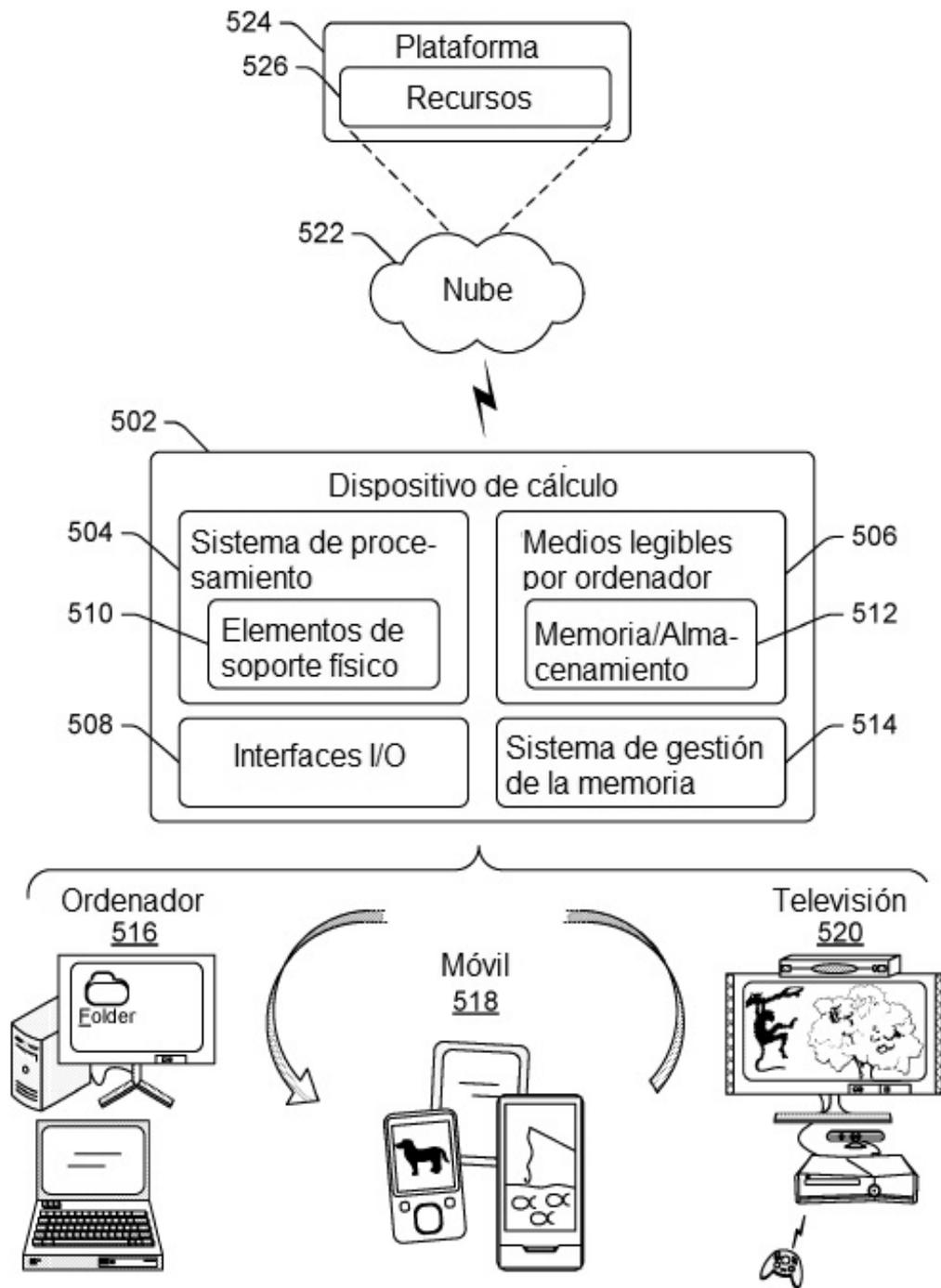


Fig. 5