

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 699 997**

51 Int. Cl.:

G06F 11/30 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **27.02.2013** E 13156871 (9)

97 Fecha y número de publicación de la concesión europea: **05.09.2018** EP 2634699

54 Título: **Vigilancia de aplicaciones**

30 Prioridad:

28.02.2012 IN MU05372012

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

13.02.2019

73 Titular/es:

**TATA CONSULTANCY SERVICES LIMITED
(100.0%)**

**Nirmal Building 9th Floor Nariman Point Mumbai
400 021
Maharashtra, IN**

72 Inventor/es:

**DASH, HEMANTA KUMAR y
MOHANTY, SUBHASISH**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 699 997 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

Vigilancia de aplicaciones

Campo técnico

5 La presente materia se refiere, en general, a aplicaciones de software y, en particular, a la vigilancia de aplicaciones de software.

Antecedentes

10 Una aplicación, también referida como una aplicación de software, es un software ejecutable que realiza tareas específicas. En un cierto instante, puede haber varios usuarios ejecutando una aplicación en particular. Además, el uso de una aplicación por número de usuarios puede obstaculizar el desempeño de la aplicación. Una aplicación puede supervisarse para obtener detalles relacionados con el rendimiento de una aplicación. Por ejemplo, una aplicación puede supervisarse para comprobar escalabilidad, capacidad de respuesta, requisitos de calidad de servicio (QoS) de una aplicación.

15 La vigilancia de aplicaciones puede ayudar a un administrador a detectar puntos de cuello de botella y responder a las anomalías antes de que el rendimiento de la aplicación se vea degradado aún más. Esto se traduce en la reducción del tiempo de inactividad del servidor para el mantenimiento de la aplicación, lo que conduce además a un mejor rendimiento de la aplicación.

20 Muchas veces, durante la vigilancia de aplicaciones, debido a múltiples usuarios que acceden a la aplicación al mismo tiempo, el tiempo de respuesta de la aplicación puede aumentar. Puede ser posible que la transferencia de un gran volumen de datos directamente de la aplicación a una base de datos pueda dar lugar a pérdidas de memoria. Además, el procedimiento de vigilancia puede obstaculizar el desempeño de la aplicación.

Técnica anterior

25 El documento US 6.658.416 B1 divulga un aparato y un procedimiento para catalogar datos simbólicos para su uso en el análisis del rendimiento de programas informáticos se proporciona. El aparato y procedimiento almacenan datos simbólicos para para los módulos cargados durante o poco después de un rastreo de rendimiento y utilizan los datos simbólicos almacenados cuando se realiza un análisis del rendimiento en un momento posterior. Un archivo de símbolo combinado se genera por un programa informático, o aplicación, bajo rastreo. El archivo de símbolo combinado contiene información útil para realizar la resolución simbólica de información de dirección en archivos de rastreo para cada instancia de un módulo. Durante el procesamiento posterior de la información de rastreo generada por un rastreo de rendimiento de un programa informático, la información simbólica almacenada en el archivo de símbolo combinado se compara con la información de rastreo almacenada en el archivo de rastreo. La información simbólica correcta en el archivo de símbolo combinado para los módulos cargados se identifica basándose en varios criterios de validación. La información simbólica correcta para los módulos cargados puede entonces almacenarse como una base de datos indexada que se utiliza para resolver la información de dirección en información simbólica correspondiente cuando se proporciona la información de rastreo a una pantalla para su uso por un usuario.

Sumario

35 Este sumario se proporciona para introducir los conceptos relacionados con un procedimiento para la vigilancia de aplicaciones y los conceptos se describen más adelante en la descripción detallada. Este sumario no tiene por objeto identificar las características esenciales de la materia reivindicada ni se pretende para su uso en la determinación o limitación del alcance de la materia reivindicada.

40 En una realización del presente objeto, se describe el uno o más procedimientos y el uno o más sistemas para la vigilancia del rendimiento de una aplicación de software. En una implementación, se obtienen los datos de vigilancia relacionados con uno o más componentes de una aplicación de software. Se obtienen los datos de vigilancia basándose en las preferencias del usuario. Los datos de vigilancia se pueden transferir a una memoria de vigilancia intermedia después de un intervalo de tiempo primero predeterminado. Los datos de vigilancia se transfieren además de la memoria de vigilancia intermedia a una base de datos después de un segundo intervalo de tiempo predeterminado. La transferencia de datos de vigilancia de la memoria de vigilancia intermedia a la base de datos es asíncrona con respecto al procedimiento de ejecución de la aplicación.

Breve descripción de los dibujos

50 La descripción detallada se proporciona con referencia a las figuras adjuntas. En las figuras, el uno o más dígitos más a la izquierda de un número de referencia identifica la figura en la que aparece por primera vez el número de referencia. Los mismos números se utilizan en todos los dibujos para hacer referencia a características y componentes iguales.

La Figura 1 ilustra un entorno de red que implementa un sistema de vigilancia de aplicaciones, de acuerdo con una implementación del presente objeto.

La Figura 2 ilustra un procedimiento para la vigilancia de una aplicación, de acuerdo con una implementación del presente objeto.

Descripción detallada

5 El presente objeto se refiere a la vigilancia de aplicaciones de software, también referidas como una aplicación, en particular, a la vigilancia del rendimiento de una aplicación Java™. Una aplicación puede incluir varios módulos y cada módulo puede realizar una tarea predefinida. En un instante particular, puede haber múltiples módulos que pueden invocarse por uno o más usuarios. Por ejemplo, el número de veces que se invoca un módulo, es decir, los recuentos de invocación pueden basarse en la popularidad o importancia de un módulo. La ejecución simultánea de los módulos conduce, a menudo, a una estructura operativa compleja de la aplicación. Para recopilar detalles
10 relativos a la eficacia, el rendimiento y otra información, se puede realizar la vigilancia de aplicaciones. Por ejemplo, la vigilancia de aplicaciones puede proporcionar ayuda en la identificación de módulos de cuello de botella y anomalías asociadas con la ejecución de tales módulos. Además, la identificación oportuna de estos cuellos de botella puede proporcionar menor tiempo de inactividad por mantenimiento y mejora de la disponibilidad de la aplicación, puesto que las anomalías asociadas a la aplicación pueden abordarse antes de que comiencen a afectar al rendimiento de la aplicación.
15

Normalmente, los datos recogidos durante la vigilancia de aplicaciones se pueden transferir a un dispositivo de almacenamiento permanente, tal como una base de datos. La comunicación directa de la función de vigilancia se llama junto con la función de ejecución de la aplicación que se llama con la base de datos puede aumentar el tráfico de red de la base de datos. Por lo general, durante la ejecución de la aplicación, el tiempo tomado por el
20 procedimiento de vigilancia puede sumarse al tiempo de respuesta de la aplicación. De esta manera resulta en un aumento no deseable en el tiempo de respuesta para el procedimiento de ejecución de la aplicación. Además, en ciertos casos, la actualización de la base de datos puede aumentar innecesariamente el tráfico de red lo que da como resultado un pobre rendimiento de la aplicación. Además, en ciertas técnicas, los datos de vigilancia pueden no almacenarse y pueden proporcionarse directamente a un administrador para los fines de vigilancia. Por tanto, en tales casos, los datos de vigilancia pueden no estar disponibles para futuras referencias. Además, muchas veces, el
25 procedimiento de vigilancia puede integrarse con el procedimiento de solicitud, lo que puede conducir a un pobre rendimiento de la aplicación puesto que el procedimiento de vigilancia puede dificultar el procedimiento de ejecución de la aplicación.

De acuerdo con una realización del presente objeto, uno o más sistemas y uno o más procedimientos para la
30 vigilancia de una aplicación se describen en la presente memoria. Además, las aplicaciones pueden ejecutarse como varios servidores de aplicaciones en un entorno de clúster. En una forma ejemplar de la invención, una aplicación puede incluir múltiples componentes, que pueden definirse en cualquier nivel granular. Por ejemplo, un componente se puede definir a un nivel de módulo, donde cada módulo realiza un procedimiento de negociación separado. Por ejemplo, una aplicación puede incluir la inscripción y gestión de proyectos como sus dos módulos.

35 Cada módulo puede comprender además múltiples sub-módulos o procedimientos de negociación, donde cada uno de tales procedimientos de negociación puede realizar una tarea específica dentro de ese módulo. Considerando el ejemplo mencionado anteriormente, la inscripción puede definir aún más su submódulo como la inscripción doméstica y la inscripción de préstamo. La inscripción de préstamo puede incluir, además, puede definir un préstamo personal, préstamo para la educación, préstamo de coche, etc., como sub-módulos adicionales.

40 El presente objeto proporciona datos relacionados con el rendimiento y la ejecución de la aplicación que ocurren en el nivel de componente. En una implementación, se pueden obtener datos relacionados con el rendimiento y detalles de ejecución de una aplicación. Tales datos pueden referirse en lo sucesivo como datos de vigilancia. Los datos de vigilancia pueden incluir detalles relativos para ralentizar el acceso a la aplicación, el tiempo de respuesta, los recuentos de invocación, consultas, etc. Además, los datos de vigilancia obtenidos pueden corresponder a los
45 componentes diana de la aplicación. La selección de los componentes diana se puede realizar basándose en las preferencias del usuario predefinidas. Tras la obtención, los datos de vigilancia se pueden transferir a una memoria de vigilancia intermedia para su almacenamiento temporal en una base periódica. Además, los datos de vigilancia se pueden transferir además a la base de datos después de un intervalo de tiempo predeterminado. En una implementación, el intervalo de tiempo predeterminado se puede configurar. Además, el procedimiento de transferencia de la memoria de vigilancia intermedia a la base de datos puede ser asíncrono con respecto al
50 procedimiento de ejecución de la aplicación. La transferencia asíncrona asegura que el procedimiento de vigilancia no obstaculice el rendimiento de las aplicaciones. Tras la transferencia exitosa de los datos de la memoria de vigilancia intermedia a la base de datos, la memoria de vigilancia intermedia se puede descargar para recibir los datos de vigilancia que corresponden al siguiente ciclo de vigilancia.

55 Además, los datos de vigilancia recogidos en la base de datos se mantienen en la base de datos y pueden buscarse por los usuarios como cuando se requiera. Además, la vigilancia de los datos puede ayudar a un usuario en la identificación de puntos de cuello de botella en la aplicación y de anomalías en el funcionamiento de la aplicación. Además, la vigilancia de aplicaciones se realiza de manera que no interfiera con la ejecución de la aplicación, que no solo resulta en un mejor rendimiento de la aplicación, sino que también asegura que se reciban datos
60 sustancialmente precisos relacionados con el rendimiento de la aplicación.

En una forma ejemplar de la invención, el procedimiento de vigilancia puede ser configurable en tiempo de ejecución. Además, el inicio y la terminación del procedimiento de vigilancia pueden ser posibles durante la ejecución de la aplicación sin necesidad de reiniciar un servidor de aplicaciones que aloja la aplicación de software. En una implementación, el inicio y terminación se pueden controlar a través de un indicador de propiedad (verdadero/falso) definido en las propiedades de vigilancia de archivos configurados en el servidor de aplicaciones. En un ejemplo, el valor 'verdadero' del indicador de la propiedad puede significar que el procedimiento de vigilancia está activado y el valor 'falso' puede significar que el procedimiento de vigilancia está desactivado. Por lo tanto, en función del valor (verdadero/falso) definido por el indicador de propiedad, el procedimiento de vigilancia se puede iniciar, continuar o suspenderse. Además, la aplicación se puede vigilar a un nivel de aplicación, nivel de base de datos, o ambos. Mientras se vigila a nivel de aplicación, los parámetros de rendimiento, tales como el recuento de invocación y el tiempo de respuesta de una solicitud del usuario pueden recogerse. Mientras se vigila a nivel de base de datos, una consulta de base de datos activada en respuesta a una transacción correspondiente a la solicitud de usuario puede recogerse.

Los procedimientos y sistemas anteriores se describen con más detalle junto con las siguientes figuras. Cabe señalar que la descripción y las figuras ilustran meramente los principios del presente objeto. Por tanto, se apreciará que los expertos en la técnica serán capaces de concebir diversas disposiciones que, aunque no se describen explícitamente o se muestran en la presente memoria, incorporan los principios del presente objeto y se incluyen dentro de su espíritu y alcance. Además, todos los ejemplos citados en esta memoria pretenden expresarse principalmente solo con fines pedagógicos para ayudar al lector en la comprensión de los principios del presente objeto y los conceptos aportados por el uno o más inventores a la promoción de la técnica, y se deben interpretar como no teniendo ninguna limitación a tales ejemplos y condiciones específicamente mencionados.

Si bien los aspectos de los sistemas y procedimientos descritos para la vigilancia de aplicaciones pueden implementarse en cualquier número de diferentes sistemas, entornos y/o configuraciones, los ejemplos se describen en el contexto de la una o más siguientes arquitecturas del sistema.

La Figura 1 ilustra un entorno 100 de red que implementa un sistema 102 de vigilancia de aplicaciones, de acuerdo con una forma ejemplar de la invención. El sistema 102 de vigilancia de aplicaciones puede proporcionarse como un servidor de aplicaciones en entorno de clúster. Aunque, un solo servidor de aplicaciones se ha ilustrado en la Figura 1, se entenderá que varios servidores de aplicaciones pueden también proporcionarse. El sistema 102 de vigilancia de aplicaciones se puede conectar a uno o más dispositivos 104-1, 104-2, ..., 104-N de usuario a través de una red 106. Con la finalidad de explicar y aclarar, los dispositivos 104-1, 104-2, ..., 104-N de usuario, se denominan en lo sucesivo colectivamente dispositivos 104 de usuario e individualmente y se refieren individualmente como dispositivo 104 de usuario.

La red 106 puede ser una red inalámbrica, una red cableada, o una combinación de las mismas. La red 106 puede también ser una red individual o una colección de muchas de tales redes individuales, interconectadas entre sí y que funcionan como una única gran red, por ejemplo, Internet o intranet. La red 106 puede implementarse como uno de los diferentes tipos de redes, como intranet, red de área local (LAN), red de área amplia (WAN), internet, y similares. La red 106 puede ser o bien una red dedicada o una red compartida, que representa una asociación de los diferentes tipos de redes que utilizan una variedad de protocolos, por ejemplo, Hypertext Transfer Protocol (HTTP), Control de Transmisión Protocol/Protocolo de Internet (TCP/IP), etc., que se comunican entre sí. Además, la red 106 puede incluir dispositivos de red, tales como conmutadores de red, concentradores y enrutadores para proporcionar un enlace entre el sistema 102 de vigilancia de aplicaciones y los dispositivos 104 de usuario. Los dispositivos de red dentro de la red 106 pueden interactuar con el sistema 102 de vigilancia de aplicaciones y el dispositivo 104 de usuario a través de los enlaces de comunicación.

El sistema 102 de vigilancia de aplicaciones y los dispositivos 104 de usuario pueden implementarse como cualquiera de una variedad de dispositivos informáticos convencionales, incluyendo, por ejemplo, servidores, un PC de sobremesa, un notebook o un ordenador portátil, un puesto de trabajo, un ordenador mainframe, y un dispositivo de Internet.

Como se ilustra, el sistema 102 de vigilancia de aplicaciones incluye uno o más procesadores 108, una o más interfaces 110 y una memoria, tal como una memoria 112, acoplada al uno o más procesadores 108. Las interfaces 110 pueden incluir una variedad de interfaces de software y hardware, por ejemplo, las interfaces para el uno o más dispositivos periféricos, tal como un teclado, un ratón, una memoria externa, y una impresora. Además, las interfaces 110 pueden permitir que el sistema 102 de vigilancia de aplicaciones se comuniquen con diferentes sistemas informáticos, tales como los dispositivos 104 de usuario. Las interfaces 110 pueden facilitar múltiples comunicaciones dentro de una amplia variedad de redes y tipos de protocolo, incluyendo las redes de cable, por ejemplo, red de área local (LAN), cable, etc., y redes inalámbricas, como LAN inalámbrica (WLAN), celular o por satélite. Para su finalidad, las interfaces 110 pueden incluir uno o más puertos para conectar los dispositivos de usuario 104 entre sí o con otro sistema informático o una base de datos, como una base 114 de datos.

El procesador 108 puede ser una sola unidad de procesamiento o un número de unidades, todas las que podrían incluir múltiples unidades informáticas. El procesador 108 puede implementarse como uno o más microprocesadores, microordenadores, microcontroladores, procesadores de señales digitales, unidades centrales

de procesamiento, máquinas de estados, circuitos lógicos, y/o cualquier dispositivo que manipule señales basándose en instrucciones operativas. Entre otras funciones, el procesador 108 se configura para buscar y ejecutar instrucciones y datos legibles por ordenador almacenados en la memoria 112.

5 Las funciones de los diversos elementos mostrados en las figuras, incluyendo cualquier bloque funcional etiquetados como "uno o más procesadores", se pueden proporcionar mediante el uso del hardware dedicado así como de hardware capaz de ejecutar el software en asociación con el software apropiado. Cuando se proporciona por un procesador, las funciones pueden proporcionarse por un único procesador dedicado, por un único procesador compartido, o por una pluralidad de procesadores individuales, algunos de los que pueden compartirse. Además, el uso explícito del término "procesador" no debe interpretarse como refiriéndose exclusivamente a un hardware capaz de ejecutar el software, y pudiendo incluir implícitamente, sin limitación, un hardware de procesador de señal digital (DSP), procesador de red, circuito integrado de aplicación específica (ASIC), Matriz de Compuerta programable en Campo (FPGA), memoria de solo lectura (ROM) para almacenar el software, memoria de acceso aleatorio (RAM), y almacenamiento no volátil

15 La memoria 112 puede incluir cualquier medio legible por ordenador conocido en la técnica incluyendo, por ejemplo, la memoria volátil, tal como memoria estática de acceso aleatorio (SRAM) y la memoria dinámica de acceso aleatorio (DRAM), y/o la memoria no volátil, tal como memoria de solo lectura (ROM), ROM programable borrrable, memorias flash, discos duros, discos ópticos, y cintas magnéticas. La memoria 112 incluye uno o más módulos de datos 116 y 118. Los módulos 116 incluyen, entre otras cosas, rutinas, programas, objetos, componentes, estructuras de datos, etc., que realizan tareas particulares o implementan tipos de datos abstractos. Los módulos 20 116 incluyen además un módulo 120 de vigilancia, un módulo 122 de transferencia de datos asíncrono, y otros módulos 124.

Los datos 118 sirven, entre otras cosas, como un repositorio para el almacenamiento de datos procesados, recibidos y generados por uno o más de los módulos 116. Los datos 118 incluyen datos 126 de vigilancia, una o más aplicaciones 127 que pueden denominarse también como una o más aplicaciones 127 de software, y otros datos 25 128. Para los fines de la ilustración, la aplicación 127 se ha mostrado como estando almacenada en los datos 118. No obstante, se entiende que la aplicación 127 puede, como alternativa, ser una parte de los módulos 116 o puede alojarse en un dispositivo de almacenamiento externo u otro dispositivo de computación. Los otros datos 128 incluyen los datos generados como resultado de la ejecución de uno o más módulos 116.

Además, la memoria 112 incluye una o más aplicaciones 127. Además, la aplicación 127 puede incluir múltiples componentes, que pueden invocarse por varios usuarios. Los componentes se pueden definir en cualquier nivel de granularidad. En un ejemplo, los componentes pueden categorizarse en un nivel de módulo. Un componente puede comprender además diversos procedimientos comerciales. Algunos de estos componentes pueden considerarse de particular importancia en el procedimiento de ejecución de la aplicación. Un usuario puede desear vigilar tales componentes que tienen una importancia particular, comparable a otros componentes. Tales componentes pueden 35 referirse como componentes diana.

En la implementación, el módulo 120 de vigilancia de la aplicación 127 se puede configurar para vigilar y proporcionar detalles relativos a la ejecución de la aplicación 127. El módulo 120 de vigilancia puede identificar los componentes diana de la aplicación 127, basándose en las preferencias del usuario. En un ejemplo, los usuarios que deseen vigilar la aplicación 127 pueden definir sus preferencias en tiempo de ejecución. Además, el módulo 120 de vigilancia pueden definir los puntos de ejecución en la aplicación 127. Los puntos de ejecución pueden considerarse como un grupo de cortes de punto y su ejecución. Los puntos de ejecución pueden ser los componentes diana que deben vigilarse. En un ejemplo, el módulo 120 de vigilancia se puede definir utilizando programación orientada a aspectos. Por tanto, el módulo 120 de vigilancia puede determinar los parámetros de rendimiento correspondientes a cada uno de los componentes diana de la aplicación 127 y almacenarse en los datos 40 126 de vigilancia. En un ejemplo, los parámetros de rendimiento pueden ser recuentos de invocación, tiempo de respuesta de consulta de base de datos, etc. Una solicitud de invocar uno o más componentes de una aplicación 127 puede referirse como solicitud de negociación. Se entenderá que cada solicitud de negociación puede invocar múltiples componentes y, por lo tanto, puede tener varias transacciones. Para cada transacción correspondiente a una solicitud de negociación, se pueden obtener los parámetros de rendimiento.

Un recuento invocación de un componente puede entenderse como el número de veces que el componente se invoca. Un contador puede mantener el registro de los recuentos de invocación que corresponden a un componente. El contador puede llevarse cada vez que hay un golpe para el componente particular. El recuento de invocación puede definir también la popularidad de ese componente particular entre los usuarios, es decir, mientras mayor es el recuento de invocación de un componente, mayor es su popularidad. Muchas veces, estos componentes populares se convierten en los componentes de cuello de botella de una aplicación 127 y mejorando el rendimiento de tales componentes se puede mejorar el rendimiento general de la aplicación 127. 55

Una vez que un componente particular es invocado por un usuario, la aplicación 127 puede tomar algún tiempo en generar una respuesta o salida requerida en respuesta a tal invocación. El tiempo empleado por la aplicación 127 para generar la salida correspondiente a una solicitud de usuario se conoce como el tiempo de respuesta para la solicitud de usuario. Basándose en la solicitud o entrada de usuario, se invoca un componente particular. Además, el 60

sistema 102 de vigilancia de aplicaciones puede disparar al menos una consulta en respuesta a la transacción para obtener la salida requerida.

El módulo 120 de vigilancia puede proporcionar los datos 126 de vigilancia asociados con los componentes diana a un dispositivo de almacenamiento temporal, tal como una memoria 130 de vigilancia intermedia. Se entenderá que los datos 126 de vigilancia corresponderán a un ciclo de vigilancia anterior, donde cada ciclo es del primer intervalo de tiempo predeterminado. En un ejemplo, la estructura de datos que se puede almacenar en la memoria 130 de vigilancia intermedia se puede predefinir por un usuario. Además, en un ejemplo, la memoria de vigilancia intermedia puede ser una memoria rápida, tal como la memoria caché. Por lo tanto, los datos 126 de vigilancia se pueden proporcionar de manera eficaz en la memoria 130 de vigilancia intermedia, con el consiguiente ahorro en tiempo y recursos informáticos.

La memoria 130 de vigilancia intermedia puede almacenar los datos 126 de vigilancia para un segundo intervalo de tiempo predeterminado. De la memoria 130 de vigilancia intermedia, el módulo 122 de transferencia de datos asíncrono puede transferir los datos 126 de vigilancia a la base 114 de datos después de cada segundo intervalo de tiempo predeterminado. Los datos 126 de vigilancia de la memoria 130 de vigilancia intermedia se transfieren a la base 114 de datos de manera que la ejecución de la aplicación 127 es independiente del procedimiento de vigilancia. Por lo tanto, el procedimiento de vigilancia no dificulta la ejecución de la aplicación 127. El trabajo independiente se puede asegurar puesto que la red para la transferencia de los datos 126 de vigilancia de la memoria 130 de vigilancia intermedia a la base 114 de datos está separada de aquella en la que la aplicación 127 se está ejecutando.

Además, el módulo 122 de transferencia de datos asíncrono puede eliminar los datos 126 de vigilancia almacenados de la memoria 130 de vigilancia intermedia una vez que un criterio de actualización de la base de datos es conocido. El criterio de actualización de la base de datos puede ser al menos uno de completar con éxito el ciclo de transferencia anterior de la memoria 130 de vigilancia intermedia a la base 114 de datos y la expiración de un tercer intervalo de tiempo predeterminado. Esto se traduce en el uso eficaz de la memoria intermedia y en la transferencia de datos sin pérdidas. La descarga de los datos 126 de vigilancia de la memoria 130 de vigilancia intermedia tras el éxito de la transferencia de datos a la base 114 de datos minimiza las posibilidades de pérdida de datos durante la transferencia y evita pérdidas de memoria. Además, se entenderá que el módulo 120 de vigilancia puede sincronizarse con el módulo 122 de transferencia de datos asíncrono de manera que la memoria 130 de vigilancia intermedia se actualice y descargue en consecuencia.

En una forma ejemplar de la invención, el primer segundo, y tercer intervalos de tiempo predeterminados pueden predefinirse por un usuario y se pueden restablecer basándose en los requisitos y los volúmenes de datos. El primer segundo, y tercer intervalos de tiempo predeterminados pueden ser iguales o diferentes dependiendo de la preferencia del usuario. La duración de los intervalos de tiempo predeterminados se puede cambiar en tiempo de ejecución lo que permite al usuario controlar el uso de memoria con la memoria 130 de vigilancia intermedia. Los intervalos de tiempo se pueden definir de manera óptima para proporcionar un procedimiento de vigilancia eficaz.

En una forma ejemplar de la invención, el módulo 122 de transferencia de datos asíncrono además se puede configurar para comprobar periódicamente el valor definido para un indicador de propiedad (Verdadero/Falso) en la vigilancia de archivos de propiedades, y controlar el inicio y la terminación del procedimiento de vigilancia basado en el estado. Por ejemplo, si el valor del indicador de propiedad cambia de 'Falso' a 'Verdadero', el módulo 122 de transferencia de datos asíncrono puede descargar los datos 126 de vigilancia que corresponden al ciclo de transferencia de datos anterior de la memoria 130 de vigilancia intermedia, cargar el archivo de propiedades para iniciar el procedimiento de vigilancia, y el procedimiento de vigilancia puede entonces proceder como se ha mencionado anteriormente. Por otra parte, cuando el valor del indicador de propiedad cambia de 'Verdadero' a 'Falso', el módulo 122 de transferencia de datos asíncrono puede de manera similar terminar el procedimiento de vigilancia continuo. Por tanto, el procedimiento de vigilancia se puede iniciar y terminar sin necesidad de reiniciar el servidor que aloja la aplicación 127.

La base 114 de datos almacena los datos 126 de vigilancia recibidos desde el módulo 122 de transferencia de datos asíncrono. La base 114 de datos puede ser un dispositivo de almacenamiento permanente y puede archivar, si es necesario. Por lo tanto, la base 114 de datos hace que los datos relativos al rendimiento de la aplicación 127 estén disponibles para futuras referencias. Por lo tanto, en lugar de transferir los datos 126 de vigilancia directamente a la base 114 de datos, se transfieren primero a la memoria 130 de vigilancia intermedia. Puesto que, la memoria 130 de vigilancia intermedia es una memoria rápida, por lo tanto, el tiempo necesario para transferir datos se reduce sustancialmente in con ello no obstaculizar el desempeño de la aplicación 127. Esto puede proporcionar un tiempo de respuesta más rápido, aumentando así el rendimiento de la aplicación 127 y tampoco sin obstaculizar la ejecución de la aplicación 127. Además, tal disposición proporciona una reducción en el tráfico de red, puesto que la base 114 de datos no está sobrecargada con un gran número de actualizaciones simultáneas.

El dispositivo 104 de usuario puede recuperar los datos 126 de vigilancia de la base 114 de datos como por solicitud con una herramienta de vigilancia gráfica, tal como JConsole™. Además, la herramienta de vigilancia gráfica puede facilitar también la búsqueda de datos en un formato requerido. El usuario puede enviar una consulta a través de dispositivo 104 de usuario correspondiente para obtener los datos de vigilancia durante un intervalo dado para una

aplicación 127 particular. La herramienta gráfica de vigilancia puede obtener los datos necesarios de la base 114 de datos y mostrarlos en el dispositivo de usuario. Por ejemplo, los datos 126 de vigilancia puede estar provisto como objetos Mbean™.

5 En una forma ejemplar de la invención, el sistema 102 de vigilancia de aplicaciones se puede configurar para analizar los datos de vigilancia basándose en reglas de análisis predefinidas y proporcionar una respuesta basada en el análisis. Por ejemplo, las reglas de análisis predefinidas pueden definir un tiempo de respuesta umbral. En dicho ejemplo, si se determina que el tiempo de respuesta para una solicitud de usuario es mayor que un tiempo de respuesta umbral; una alerta se puede proporcionar a un usuario en cuestión. En otro ejemplo, reglas de análisis predefinidas pueden describir las reglas para identificar los componentes indicadores de color rojo, por ejemplo, 10 basándose en los recuentos de invocación o tiempo de respuesta. En dicho ejemplo, el sistema 102 de vigilancia de aplicaciones puede resaltar los componentes indicadores de color rojo o el componente más popular.

15 En otra forma ejemplar de la invención, en la búsqueda de datos en la base 114 de datos, un usuario puede analizar los datos de vigilancia buscados y tomar medidas correctivas para mejorar el rendimiento de varios componentes de la aplicación 127. Esto proporciona además la mejora del rendimiento de la aplicación 127 mediante la incorporación de los cambios apropiados en la aplicación 127. Por ejemplo, los componentes populares se pueden ajustar para mejorar el rendimiento de la aplicación 127. Además, la identificación de componentes popular puede ayudar en la toma de decisiones relacionadas con la negociación, así, por ejemplo, el área de mayor crecimiento.

20 El presente objeto proporciona además la vigilancia de aplicaciones para aplicaciones cliente-servidor, así como para aplicaciones de escritorio. En las aplicaciones de escritorio, los datos 126 de vigilancia se pueden ver mediante la integración de un servidor, implementado el procedimiento de vigilancia descrito, con la aplicación de escritorio.

La Figura 2 ilustra un procedimiento 200 para vigilar de aplicaciones, de acuerdo con una implementación del presente objeto. El procedimiento 200 puede implementarse en una variedad de sistemas informáticos en varias maneras diferentes. Por ejemplo, el procedimiento 200, que se describe en la presente memoria, puede implementarse utilizando el sistema 102 de vigilancia de aplicaciones.

25 El procedimiento 200 puede describirse, total o parcialmente, en el contexto general de instrucciones ejecutables por ordenador. Por lo general, las instrucciones ejecutables por ordenador pueden incluir rutinas, programas, objetos, componentes, estructuras de datos, procedimientos, módulos, funciones, etc., que realizan funciones particulares o implementan tipos de datos abstractos. Un experto en la materia reconocerá fácilmente que las etapas del procedimiento pueden realizarse por ordenadores programados. En la presente memoria, algunos ejemplos 30 pretenden cubrir también los dispositivos de almacenamiento del programa, por ejemplo, medios de almacenamiento de datos digitales, que son legibles por máquina u ordenador y codificar programas de instrucciones ejecutables por máquina o ejecutables por ordenador, en los que dichas instrucciones realizan todas o algunas de las etapas del procedimiento 200 descrito.

35 Haciendo referencia a la Figura 2, en el bloque 202, una o más solicitudes que invocan los componentes diana correspondientes a una aplicación, tales como la aplicación 127, se reciben de uno o más dispositivos de usuario, tales como los dispositivos 104 de usuario. Las solicitudes pueden invocar uno o más componentes dianas de la aplicación 127. Un componente diana puede ser un componente que se tiene que vigilar y tales componentes diana se pueden seleccionar por un usuario en tiempo de ejecución o antes de la ejecución de la aplicación 127. Después de la invocación de al menos uno de los componentes diana, el procedimiento 200 se bifurca al bloque 204.

40 En el bloque 204, los datos de vigilancia correspondientes a los componentes diana invocados se obtienen. Los datos 126 de vigilancia pueden incluir parámetros de rendimiento asociados con cada uno de los componentes diana. Los parámetros de rendimiento pueden ser el tiempo de respuesta correspondiente a una solicitud de usuario asociada con un componente diana, el recuento de invocación de un componente diana, las consultas activadas en respuesta a una transacción, etc.

45 En el bloque 206, los datos 126 de vigilancia obtenidos se transfieren a una memoria de vigilancia intermedia, tal como la memoria 130 de vigilancia intermedia después de un primer intervalo de tiempo predeterminado. Por lo tanto, los datos 126 de vigilancia se transfieren a la memoria 130 de vigilancia intermedia periódicamente. En una implementación, el módulo 120 de vigilancia transfiere los datos 126 de vigilancia a la memoria 130 de vigilancia intermedia.

50 En el bloque 208, se determina si un segundo intervalo de tiempo predeterminado ha terminado. En una implementación, el módulo 122 de transferencia de datos asíncrono determina si el segundo intervalo de tiempo predeterminado ha terminado. Si se determina que el segundo intervalo de tiempo predeterminado ya ha terminado, (ramificación "Sí" del bloque 208), el procedimiento 200 se bifurca al bloque 210. En el bloque 210, los datos 126 de vigilancia de la memoria 130 de vigilancia intermedia se transfieren de la memoria 130 de vigilancia intermedia a una 55 base de datos, tal como la base 114 de datos. Por lo tanto, la base 114 de datos se actualiza con los últimos datos de vigilancia periódicamente. Sin embargo, si se determina que el segundo intervalo de tiempo predeterminado no ha terminado, (ramificación "No" del bloque 208), el procedimiento 200 se ramifica de vuelta al bloque 208.

En el bloque 212, se determina si se cumple un criterio de actualización de la base de datos. Si se determina que se

ES 2 699 997 T3

cumple el criterio de actualización de la base de datos, (rama "Sí" del bloque 212), el procedimiento 200 se mueve al bloque 214. En una implementación, el módulo 122 de transferencia de datos asíncrono determina si se cumple el criterio de actualización de la base de datos.

- 5 En el bloque 214, los datos 126 de vigilancia se pueden eliminar de la memoria 130 de vigilancia intermedia, cuando se cumple el criterio de actualización de la base de datos. En un ejemplo, el módulo 122 de transferencia de datos asíncrono puede vaciar la memoria 130 de vigilancia intermedia después de la transferencia exitosa de los datos 126 de vigilancia, para garantizar la transmisión de datos sin pérdidas. En otro ejemplo, el módulo 122 de transferencia de datos asíncrono puede vaciar la memoria 130 de vigilancia intermedia después del transcurso de un tercer intervalo de tiempo predeterminado y transferir los datos con éxito.
- 10 Con referencia de nuevo al bloque 212, si se determina que el criterio de actualización de la base de datos no se cumple (rama "No" del bloque 212), el procedimiento 200 se ramifica de nuevo al bloque 212 y la transferencia de los datos 126 de vigilancia de la memoria 130 de vigilancia intermedia se continúa.

REIVINDICACIONES

1. Un procedimiento implementado por un ordenador para la vigilancia de una aplicación (127) de software, comprendiendo el procedimiento:

5 obtener datos (126) de vigilancia pertenecientes a uno o más componentes diana de la aplicación (127) de software, basándose en las preferencias de un usuario, en el que los datos (126) de vigilancia comprenden datos relativos a al menos un parámetro de rendimiento de la aplicación (127) de software en al menos uno de un nivel de aplicación y un nivel de base de datos;
 proporcionar los datos (126) de vigilancia a una memoria (130) de vigilancia intermedia después de un primer intervalo de tiempo predeterminado, en el que la memoria de vigilancia intermedia es una memoria rápida; y
 10 transferir los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a una base (114) de datos después de un segundo intervalo de tiempo predeterminado, en el que una red para la transferencia de los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a la base (114) de datos está separada de una red en la que se ejecuta la aplicación (127) de software para asegurar que la transferencia de los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a la base (114) de datos sea independiente de la ejecución de la aplicación (127) de software;
 15 determinar si se cumple un criterio de actualización de la base de datos; y
 descargar los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia basándose en el criterio de actualización de la base de datos, cuando se cumple el criterio de actualización de la base de datos, en el que la actualización de la base de datos comprende al menos uno de completar con éxito el ciclo de transferencia anterior de la memoria (130) de vigilancia intermedia a la base (114) de datos y la expiración de un tercer intervalo de tiempo predeterminado, y
 20 continuar la transferencia de los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a la base (114) de datos, cuando no se cumple el criterio de actualización de la base de datos.

25 2. El procedimiento implementado por ordenador de acuerdo con la reivindicación 1, en el que el al menos un parámetro (126) de rendimiento incluye al menos uno de un tiempo de respuesta, una consulta de base de datos, y un recuento de invocación correspondiente a cada uno del uno o más componentes diana.

30 3. El procedimiento implementado por ordenador de acuerdo con la reivindicación 1, en el que el procedimiento comprende además definir uno o más puntos de ejecución en la aplicación (127) de software para la vigilancia, basándose en las preferencias del usuario, en el que cada uno del uno o más puntos de ejecución corresponde a un componente diana de la aplicación (127) de software.

4. Un sistema de vigilancia de aplicaciones (102) que comprende:

un procesador (108); y
 una memoria (112) acoplada al procesador (108), comprendiendo la memoria (112):
 un módulo de vigilancia (120) configurado para:

35 obtener datos (126) de vigilancia pertenecientes a uno o más componentes diana de la aplicación (127) de software, basándose en las preferencias de un usuario, en el que los datos (126) de vigilancia comprenden los datos relativos a al menos un parámetro de rendimiento de la aplicación (127) de software en al menos uno de un nivel de aplicación y un nivel de base de datos;
 proporcionar los datos (126) de vigilancia correspondientes al uno o más componentes diana de la aplicación (127) de software a una memoria (130) de vigilancia intermedia después de un primer intervalo de tiempo predeterminado, en el que la memoria de vigilancia intermedia es una memoria rápida; y
 40 un módulo (122) de transferencia de datos asíncrono configurado para
 transferir los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a una base (114) de datos después de un segundo intervalo de tiempo predeterminado, en el que una red para la transferencia de los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a la base (114) de datos está separada de una red en la que se ejecuta la aplicación (127) de software para asegurar que la transferencia de los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a la base (114) de datos sea independiente de la ejecución de la aplicación (127) de software;
 45 determinar si se cumple un criterio de actualización de la base de datos; y
 50 descargar los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia basándose en el criterio de actualización de la base de datos, cuando se cumple el criterio de actualización de la base de datos, en el que la actualización de la base de datos comprende al menos uno de completar con éxito el ciclo de transferencia anterior de la memoria (130) de vigilancia intermedia a la base (114) de datos y la expiración de un tercer intervalo de tiempo predeterminado, y
 55 continuar la transferencia de los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a la base (114) de datos cuando no se cumple el criterio de actualización de la base de datos.

5. El sistema de vigilancia de aplicaciones (102) de acuerdo con la reivindicación 4, en el que el módulo de vigilancia (120) se configura además para identificar el uno o más componentes diana basándose en las preferencias de un usuario.

6. El sistema de vigilancia de aplicaciones (102) de acuerdo con la reivindicación 4, en el que el módulo (122) de transferencia de datos asíncrono se configura además para:

5 comprobar periódicamente un estado de un indicador de propiedad correspondiente a la aplicación (127) de software; y
vigilar el inicio y una terminación de un procedimiento de vigilancia para la aplicación (127) de software basándose en el estado.

7. El sistema de vigilancia de aplicaciones (102) de acuerdo con la reivindicación 4, en el que el sistema de vigilancia de aplicaciones (102) se configura para analizar los datos (126) de vigilancia basándose en reglas de análisis predefinidas y proporcionar una respuesta basada en el análisis.

10 8. El sistema de vigilancia de aplicaciones (102) de acuerdo con la reivindicación 7, en el que las reglas de análisis predefinidas incluyen al menos una de:

reglas que definen un tiempo de respuesta umbral para una consulta del usuario; y
reglas que identifican componentes indicadores de color rojo de la aplicación (127) de software.

15 9. El sistema de vigilancia de aplicaciones (102) de acuerdo con la reivindicación 4, en el que la memoria (130) de vigilancia intermedia es un dispositivo de almacenamiento temporal y la base de datos es un dispositivo de almacenamiento permanente.

10. Un medio legible por ordenador que tiene incorporado en su interior un programa informático que comprende instrucciones que, cuando el programa es ejecutado por un ordenador, hace que el ordenador realice las etapas de:

20 obtener datos (126) de vigilancia pertenecientes a uno o más componentes diana de la aplicación (127) de software, basados en las preferencias de un usuario, en el que los datos (126) de vigilancia comprenden datos relativos a al menos un parámetro de rendimiento de la aplicación (127) de software en al menos uno de un nivel de aplicación y un nivel de base de datos;

proporcionar los datos (126) de vigilancia en una memoria (130) de vigilancia intermedia después de un primer intervalo de tiempo predeterminado, en el que la memoria de vigilancia intermedia es una memoria rápida; y

25 transferir los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a una base (114) de datos después de un segundo intervalo de tiempo predeterminado, en el que una red para la transferencia de los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a la base (114) de datos se separa de una red en la que se ejecuta la aplicación (127) de software para asegurar que la transferencia de los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a la base (114) de datos sea independiente de la ejecución de la aplicación de software (127);

30 determinar si se cumple un criterio de actualización de la base de datos; y
descargar los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia basándose en el criterio de actualización de la base de datos, cuando se cumple el criterio de actualización de la base de datos, en el que la actualización de la base de datos comprende al menos uno de completar con éxito el ciclo de transferencia anterior de la memoria (130) de vigilancia intermedia a la base (114) de datos y la expiración de un tercer

35 intervalo de tiempo predeterminado, y
continuar la transferencia de los datos (126) de vigilancia de la memoria (130) de vigilancia intermedia a la base (114) de datos cuando no se cumple el criterio de actualización de la base de datos.

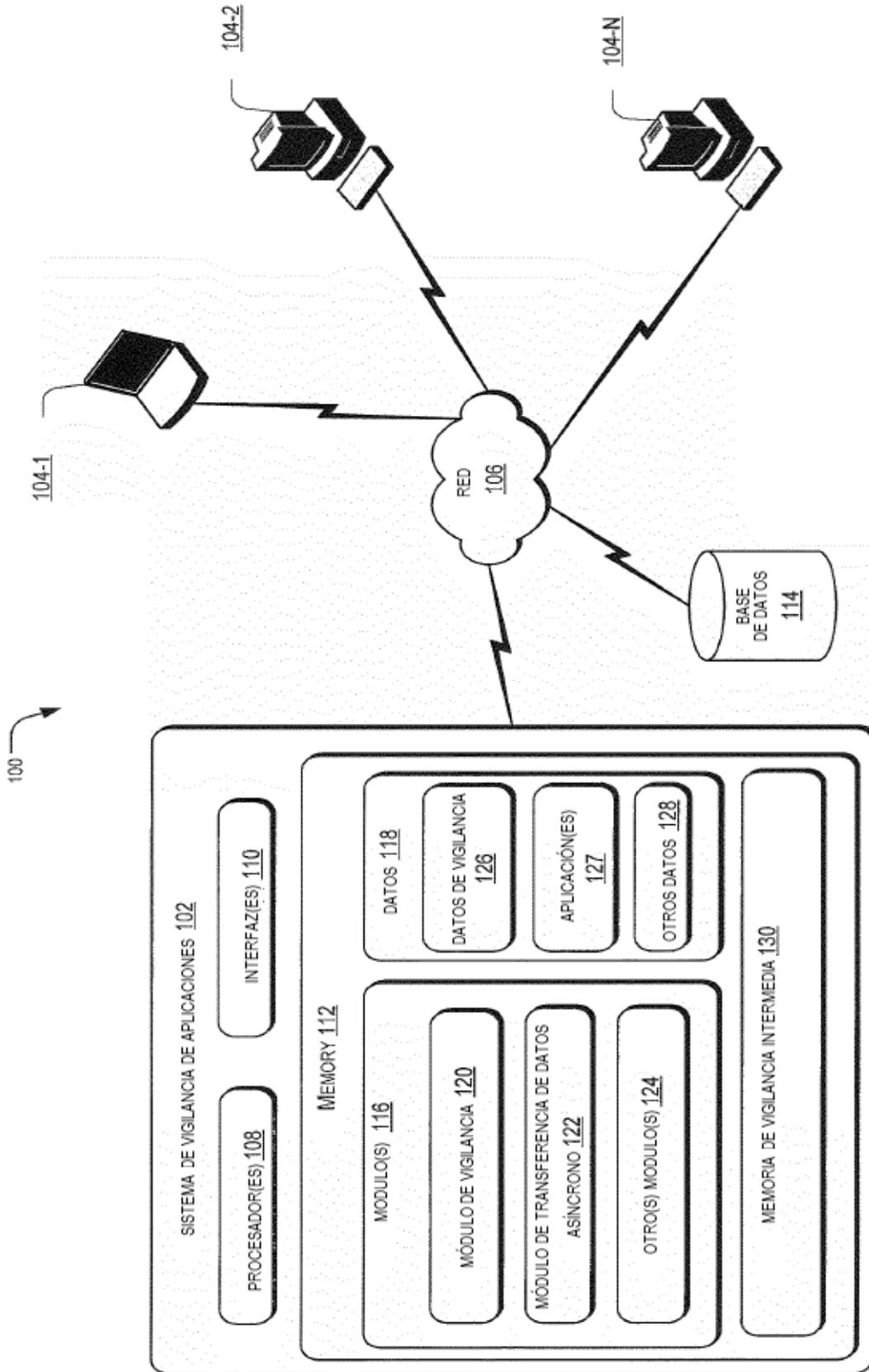


Fig. 1

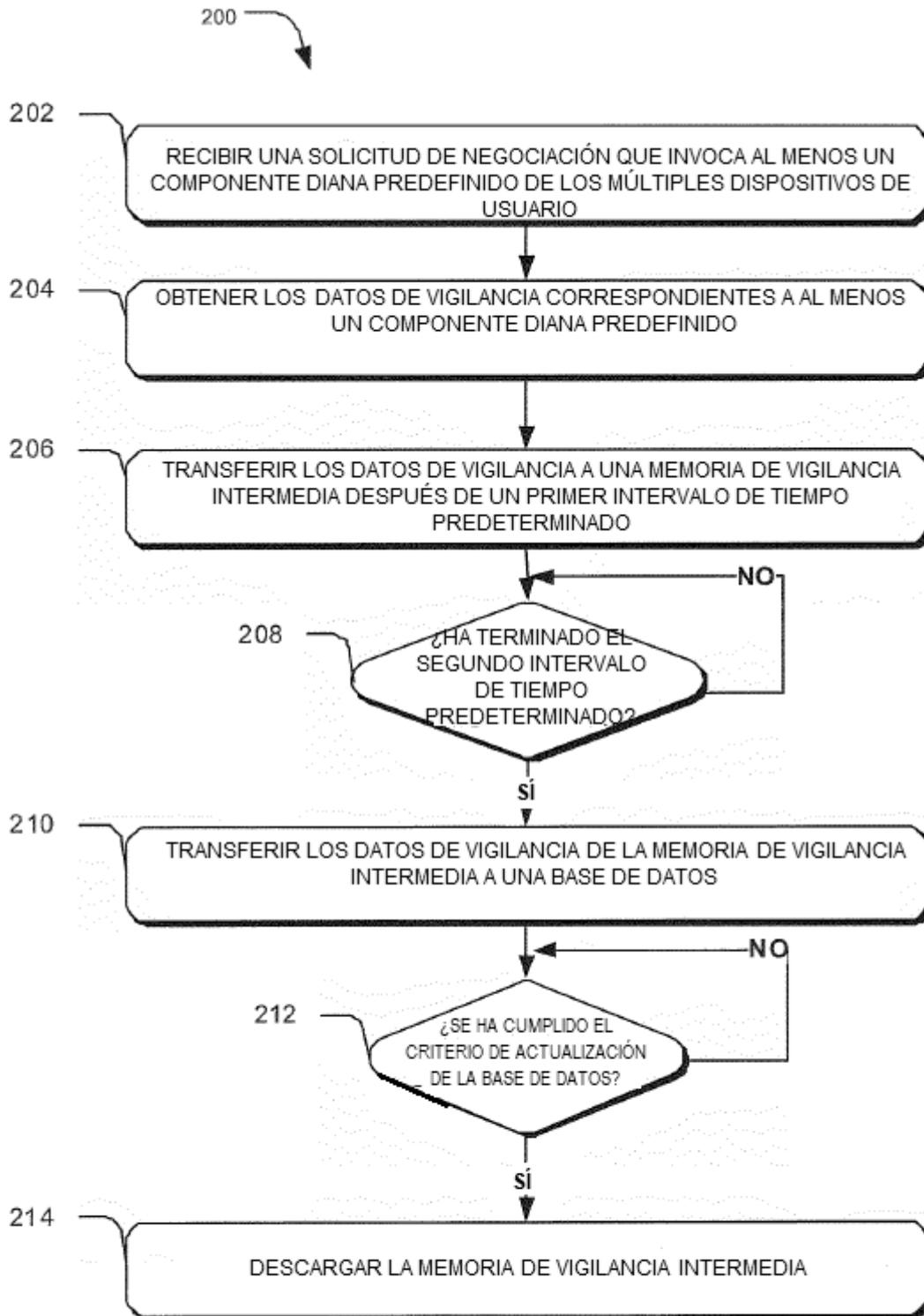


Fig. 2