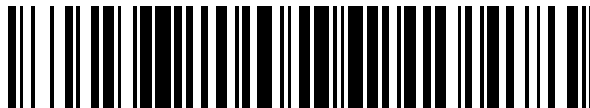


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 587 603**

51 Int. Cl.:

G06F 11/14 (2006.01)

G06F 11/20 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **26.06.2014 E 14174056 (3)**

97 Fecha y número de publicación de la concesión europea: **01.06.2016 EP 2819015**

54 Título: **Procedimiento, terminal y servidor para sincronizar información espejo de un terminal**

30 Prioridad:

28.06.2013 CN 201310268703

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

25.10.2016

73 Titular/es:

**HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building, Bantian,
Longgang District
Shenzhen, Guangdong 518129, CN**

72 Inventor/es:

**TAN, CHENG;
WU, XIAOXIN;
XIA, YUBIN y
CHEN, HAIBO**

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 587 603 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

Procedimiento, terminal y servidor para sincronizar información espejo de un terminal

5 Campo técnico

La presente invención se refiere al campo de las tecnologías de comunicación en red y, en particular, a un procedimiento, un terminal y un servidor para sincronizar información espejo de un terminal.

10 Antecedentes

A medida que se desarrollan los terminales inteligentes, es posible ejecutar más aplicaciones de software en los mismos, de modo que el usuario de un terminal puede enviar un correo electrónico, acceder a una red social, realizar pagos a través de un terminal y ejecutar varios juegos de aplicaciones usando el terminal inteligente. En un proceso mediante el cual un terminal inteligente ejecuta software de aplicación se genera una gran cantidad de datos de usuario, incluidos datos de estado generados en tiempo real durante el tiempo de ejecución del software de aplicación y datos de archivo almacenados en un disco tras la ejecución del software de aplicación. Estos datos están relacionados, posiblemente, con la privacidad del usuario del terminal. Cuando el terminal inteligente se extravía, los datos de usuario del terminal inteligente también se pierden en consecuencia, o cuando el terminal inteligente está infectado con software malicioso, los datos de usuario del terminal inteligente están en peligro. Por lo tanto, para proteger la totalidad de los datos almacenados en el terminal inteligente y la privacidad del usuario, el terminal inteligente puede, realizando una sincronización de información espejo con un servidor en la nube, permitir que el servidor en la nube obtenga los datos de usuario almacenados en el terminal inteligente con el fin de crear una copia de seguridad de los datos de usuario del terminal inteligente y llevar a cabo una comprobación de seguridad de los datos de usuario usando una gran cantidad de recursos informáticos del servidor en la nube. En la técnica anterior, por ejemplo en el documento US 2006/0085792 A1 y en el documento US 8291170, el terminal inteligente puede transmitir al servidor en la nube, a través de una red, los datos de archivo generados después de la ejecución del software de aplicación. Sin embargo, puesto que los datos de estado en tiempo real durante la ejecución del software de aplicación no pueden sincronizarse con este tipo de sincronización, los datos de usuario copiados en el servidor en la nube no están completos y, por consiguiente, la comprobación de seguridad basada en los datos de usuario no es precisa.

Resumen

35 Las formas de realización de la presente invención proporcionan un procedimiento según la reivindicación 1, un terminal según la reivindicación 7 y un servidor según la reivindicación 10 para sincronizar información espejo de un terminal, con el fin de resolver el problema de una copia de seguridad de datos incompleta que puede producirse con la manera existente de sincronizar información espejo de un terminal.

40 Para resolver el anterior problema técnico, las formas de realización de la presente invención dan a conocer las siguientes soluciones técnicas.

Según un primer aspecto, se proporciona un procedimiento para sincronizar información espejo de un terminal, que incluye:

45 obtener, mediante un terminal, múltiples eventos de entrada durante la ejecución de software de aplicación; agregar, mediante el terminal, los múltiples eventos de entrada para obtener un evento agregado; y transmitir, mediante el terminal, el evento agregado a un servidor; analizar, mediante el servidor, el evento agregado para obtener los múltiples eventos de entrada; 50 procesar, mediante el servidor, los múltiples eventos de entrada mediante software de aplicación ejecutado en una máquina virtual del terminal para obtener datos de usuario generados durante la ejecución del software de aplicación.

55 Con referencia al primer aspecto, en un primer modo de implementación posible del primer aspecto, antes de agregar, mediante el terminal, los múltiples eventos de entrada, el procedimiento incluye además: insertar, mediante el terminal, los eventos de entrada obtenidos en una cola de eventos según una secuencia de generación de eventos; donde, de manera correspondiente, la agregación, mediante el terminal, de los múltiples eventos de entrada para obtener un evento agregado incluye:

60 agregar, mediante el terminal, los eventos de entrada en la cola de eventos según un periodo de tiempo fijado para obtener el evento agregado.

65 Con referencia al primer aspecto o al primer modo de implementación posible del primer aspecto, en un segundo modo de implementación posible del primer aspecto, la agregación, mediante el terminal, de los múltiples eventos de entrada para obtener un evento agregado incluye:

obtener, mediante el terminal, un evento de entrada actual según una secuencia de generación de los múltiples eventos de entrada;
 generar, mediante el terminal, información de mapa de bits para cada elemento de información de atributo del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente de un evento de entrada anterior del evento de entrada actual;
 5 determinar, mediante el terminal, la información de mapa de bits y la información de atributo, que es diferente de la del evento de entrada anterior, del evento de entrada actual como datos que van a transmitirse del evento de entrada actual; y
 10 combinar, mediante el terminal tras determinarse los datos que van a transmitirse de cada evento de entrada de los múltiples eventos de entrada, los datos que van a transmitirse de los múltiples eventos de entrada para formar el evento agregado de los múltiples eventos de entrada.

Con referencia al segundo aspecto, en un primer modo de implementación posible del segundo aspecto, la recepción, mediante un servidor, de un evento agregado transmitido por un terminal incluye: recibir, mediante el servidor, un evento agregado comprimido transmitido por el terminal, donde el evento agregado comprimido es un evento obtenido por el terminal después de comprimirse el evento agregado; donde antes de analizar, mediante el servidor, el evento agregado, el procedimiento incluye además:

20 descomprimir, mediante el servidor, el evento agregado comprimido para obtener el evento agregado.

Con referencia al segundo aspecto o al primer modo de implementación posible del segundo aspecto, en un segundo modo de implementación posible del segundo aspecto, el análisis, mediante el servidor, del evento agregado para obtener los múltiples eventos de entrada incluye:

25 obtener en orden, mediante el servidor, los datos transmitidos de un evento de entrada actual del evento agregado, donde los datos transmitidos incluyen información de mapa de bits de cada elemento de información de atributo del evento de entrada actual e información de atributo, que es diferente de la de un evento de entrada anterior, del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente del evento de entrada anterior;
 30 obtener, mediante el servidor según la indicación de la información de mapa de bits, información de atributo, que es la misma que la del evento de entrada anterior, del evento de entrada actual a partir de la información de atributo del evento de entrada anterior; y
 35 combinar, mediante el servidor, la información de atributo idéntica obtenida y la información de atributo diferente incluida en los datos transmitidos para formar el evento de entrada actual.

Con referencia al segundo aspecto, al primer modo de implementación posible del segundo aspecto o al segundo modo de implementación posible del segundo aspecto, en un tercer modo de implementación posible del segundo aspecto, el procesamiento, mediante el servidor, de los múltiples eventos de entrada usando una máquina virtual incluye:

45 cuando el servidor recibe una notificación que indica que se ha producido un cambio en un estado de sistema de la máquina virtual, determinar si la información de atributo de los múltiples eventos de entrada coincide con el estado de sistema; e
 insertar, mediante el servidor, un evento de entrada correspondiente a la información de atributo que coincide con el estado de sistema en un sistema de la máquina virtual, de modo que el sistema invoque una función de procesamiento para procesar el evento de entrada.

50 Según un aspecto adicional, se proporciona un terminal, donde el terminal incluye: un procesador y una interfaz de red, donde:

el procesador está configurado para obtener múltiples eventos de entrada durante la ejecución de software de aplicación, y para agregar los múltiples eventos de entrada para obtener un evento agregado; y
 55 la interfaz de red está configurada para transmitir el evento agregado a un servidor, de modo que tras analizar el evento agregado para obtener los múltiples eventos de entrada, el servidor procesa los múltiples eventos de entrada mediante software de aplicación ejecutado en una máquina virtual del terminal, con el fin de obtener datos de usuario generados durante la ejecución del software de aplicación.

60 Con referencia al quinto aspecto, en un primer modo de implementación posible del quinto aspecto, el procesador está configurado además para insertar los eventos de entrada obtenidos en una cola de eventos según una secuencia de generación de eventos; y el procesador está configurado específicamente para agregar los eventos de entrada en la cola de eventos según un periodo de tiempo fijado para obtener el evento agregado.

65 Con referencia al quinto aspecto o al primer modo de implementación posible del quinto aspecto, en un segundo modo de implementación posible del quinto aspecto, el procesador está configurado específicamente para: obtener

un evento de entrada actual según una secuencia de generación de los múltiples eventos de entrada; generar información de mapa de bits para cada elemento de información de atributo del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente de un evento de entrada anterior del evento de entrada actual; determinar la información de mapa de bits e información de atributo, que es diferente de la del evento de entrada anterior, del evento de entrada actual como datos que van a transmitirse del evento de entrada actual; y después de determinar los datos a transmitir de cada evento de entrada de los múltiples eventos de entrada, combinar los datos a transmitir de los múltiples eventos de entrada para formar el evento agregado de los múltiples eventos de entrada.

Según un aspecto adicional, se proporciona un servidor, donde el servidor incluye una interfaz de terminal y un procesador, donde:

la interfaz de terminal está configurada para recibir un evento agregado transmitido por un terminal, donde el evento agregado es un evento agregado obtenido por el terminal agregando múltiples eventos de entrada tras obtener los múltiples eventos de entrada durante la ejecución de software de aplicación; y el procesador está configurado para analizar el evento agregado para obtener los múltiples eventos de entrada, y para procesar los múltiples eventos de entrada mediante software de aplicación ejecutado en una máquina virtual del terminal para obtener datos de usuario generados durante la ejecución del software de aplicación.

Con referencia a un aspecto adicional, en un primer modo de implementación posible del aspecto adicional, la interfaz de terminal está configurada específicamente para recibir un evento agregado comprimido transmitido por el terminal, donde el evento agregado comprimido es un evento obtenido por el terminal después de comprimir el evento agregado; y el procesador está configurado además para descomprimir el evento agregado comprimido para obtener el evento agregado.

Con referencia al aspecto adicional o al primer modo de implementación posible del sexto aspecto, en un segundo modo de implementación posible del sexto aspecto, el procesador está configurado específicamente para: obtener en orden datos transmitidos de un evento de entrada actual a partir del evento agregado, donde los datos transmitidos incluyen información de mapa de bits de cada elemento de información de atributo del evento de entrada actual e información de atributo, que es diferente de la de un evento de entrada anterior, del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente del evento de entrada anterior; obtener, según la indicación de la información de mapa de bits, información de atributo, que es la misma que la del evento de entrada anterior, del evento de entrada actual a partir de la información de atributo del evento de entrada anterior; y combinar la información de atributo idéntica obtenida y la información de atributo diferente incluida en los datos transmitidos para formar el evento de entrada actual.

Con referencia al aspecto adicional, al primer modo de implementación posible del sexto aspecto o al segundo modo de implementación posible del sexto aspecto, en un tercer modo de implementación posible del sexto aspecto, el procesador está configurado específicamente para: cuando se recibe una notificación que indica que se ha producido un cambio en un estado de sistema de una máquina virtual, determinar si la información de atributo de los múltiples eventos de entrada coincide con el estado de sistema; e insertar un evento de entrada correspondiente a la información de atributo que coincide con el estado de sistema en un sistema de la máquina virtual, de manera que el sistema invoque una función de procesamiento para procesar el evento de entrada.

En las formas de realización de la presente invención, un terminal obtiene múltiples eventos de entrada durante la ejecución de software de aplicación, agrega los múltiples eventos de entrada para obtener un evento agregado y transmite el evento agregado a un servidor; tras analizar el evento agregado para obtener los múltiples eventos de entrada, el servidor procesa los múltiples eventos de entrada mediante software de aplicación ejecutado en una máquina virtual del terminal, para obtener los datos de usuario generados durante la ejecución del software de aplicación. En las formas de realización de la presente invención, el terminal no necesita transmitir los datos de usuario al servidor, sino que transmite los eventos de entrada al servidor de una manera basada en eventos, de modo que un servidor inicia la ejecución de los eventos de entrada usando la máquina virtual del terminal con el fin de obtener los datos de usuario idénticos a los de un terminal que ejecuta el software de aplicación, garantizándose así que el servidor pueda realizar una copia de seguridad completa de los datos de usuario y mejorándose la precisión de la comprobación de seguridad en los datos de usuario. Además, debido a que el tamaño de los eventos de entrada agregados transmitidos por el terminal es pequeño, en comparación con el tamaño para transmitir directamente los datos de usuario, se utilizan menos recursos de transmisión en red y el rendimiento del propio terminal no se ve afectado, lo que garantiza la normal ejecución de otros servicios en el terminal.

Breve descripción de los dibujos

5 Para describir más claramente las soluciones técnicas de las formas de realización de la presente invención o de la técnica anterior, a continuación se exponen brevemente los dibujos adjuntos necesarios para describir las formas de realización o la técnica anterior. Un experto en la técnica puede obtener otros dibujos a partir de estos dibujos adjuntos sin llevar a cabo investigaciones adicionales.

La FIG. 1A es un diagrama de flujo de una forma de realización de un procedimiento para sincronizar información espejo de un terminal según la presente invención.

10 La FIG. 1B es un diagrama de flujo de otra forma de realización de un procedimiento para sincronizar información espejo de un terminal según la presente invención.

La FIG. 2 es un diagrama de flujo de otra forma de realización de un procedimiento para sincronizar información espejo de un terminal según la presente invención.

15 La FIG. 3A es un diagrama esquemático de una arquitectura de red que aplica una forma de realización de la presente invención.

20 La FIG. 3B es un diagrama esquemático que ilustra una agregación de eventos que aplica una forma de realización de la presente invención.

La FIG. 4 es un diagrama de bloques de una forma de realización de un aparato para sincronizar información espejo de un terminal según la presente invención.

25 La FIG. 5 es un diagrama de bloques de otra forma de realización de un aparato para sincronizar información espejo de un terminal según la presente invención.

La FIG. 6 es un diagrama de bloques de una forma de realización de un terminal según la presente invención.

30 La FIG. 7 es un diagrama de bloques de otra forma de realización de un aparato para sincronizar información espejo de un terminal según la presente invención.

La FIG. 8 es un diagrama de bloques de otra forma de realización de un aparato para sincronizar información espejo de un terminal según la presente invención.

35 La FIG. 9 es un diagrama de bloques de una forma de realización de un servidor según la presente invención.

Descripción de formas de realización

40 Con el fin de permitir que un experto en la técnica entienda mejor las soluciones técnicas de las formas de realización de la presente invención y hacer que los objetivos, características y ventajas de las formas de realización de la presente invención sean más comprensibles, a continuación se describen en detalle las soluciones técnicas de las formas de realización de la presente invención con referencia a los dibujos adjuntos.

45 Se hace referencia a la FIG. 1A, que es un diagrama de flujo de una forma de realización de un procedimiento para sincronizar información espejo de un terminal según la presente invención. En la forma de realización, un proceso de sincronización se describe en un terminal.

50 Etapa 101: Un terminal obtiene múltiples eventos de entrada durante la ejecución de software de aplicación.

En la forma de realización, el terminal puede incluir varios teléfonos inteligentes, ordenadores de tipo tableta y similares. Un software de aplicación diferente en el terminal puede iniciar un evento de entrada de un tipo diferente durante el tiempo de ejecución. Estos eventos de entrada pueden incluir un evento de entrada táctil para llevar a cabo una operación en una pantalla, un evento de entrada de posicionamiento generado por un dispositivo GPS (Sistema de Posicionamiento Global), un evento de entrada de detección generado por un sensor, etc. Por ejemplo, cuando un usuario de un terminal trata de ejecutar software de aplicación, hace clic sobre un icono del software de aplicación en una pantalla táctil del terminal, donde la acción de hacer clic sobre el icono puede abstraerse en un evento de entrada. Cuando el usuario del terminal lleva a cabo una operación en el software de aplicación iniciado, también se inician diferentes eventos de entrada según diferentes gestos de entrada en un proceso de operación.

60 Etapa 102: Agregar los múltiples eventos de entrada para obtener un evento agregado.

En la forma de realización, el terminal puede obtener un evento de entrada actual según una secuencia de generación de los múltiples eventos de entrada; generar información de mapa de bits para cada elemento de información de atributo del evento de entrada actual, donde la información de mapa de bits de cada elemento de

información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente de un evento de entrada anterior del evento de entrada actual; determinar la información de mapa de bits y la información de atributo, que es diferente de la del evento de entrada anterior, del evento de entrada actual como datos que van a transmitirse del evento de entrada actual; y después de determinar los datos a transmitir de cada evento de entrada de los múltiples eventos de entrada, combinar los datos a transmitir de los múltiples eventos de entrada para formar el evento agregado de los múltiples eventos de entrada.

Etapa 103: Transmitir el evento agregado a un servidor, de modo que tras analizar el evento agregado para obtener los múltiples eventos de entrada, el servidor procesa los múltiples eventos de entrada usando una máquina virtual que es del terminal y que está instalada en el servidor, con el fin de obtener datos de usuario generados durante la ejecución del software de aplicación.

Puede observarse que en la forma de realización, un terminal no necesita transmitir datos de usuario a un servidor, sino que transmite eventos de entrada al servidor de una manera basada en eventos, de modo que un servidor inicia la ejecución de los eventos de entrada usando una máquina virtual del terminal con el fin de obtener datos de usuario idénticos a los de un terminal que ejecuta software de aplicación, garantizándose así que el servidor pueda realizar una copia de seguridad completa de los datos de usuario y mejorándose la precisión de la comprobación de seguridad en los datos de usuario. Además, debido a que el tamaño de un evento de entrada agregado transmitido por el terminal es relativamente pequeño, en comparación con el tamaño para transmitir directamente los datos de usuario, se utilizan menos recursos de transmisión en red y el rendimiento del propio terminal no se ve afectado, lo que garantiza la normal ejecución de otros servicios en el terminal.

Se hace referencia a la FIG. 1B, que es un diagrama de flujo de otra forma de realización de un procedimiento para sincronizar información espejo de un terminal según la presente invención. En la forma de realización, un proceso de sincronización se describe en un servidor.

Etapa 111: Un servidor recibe un evento agregado transmitido por un terminal, donde el evento agregado es un evento agregado obtenido por el terminal agregando múltiples eventos de entrada después de obtener los múltiples eventos de entrada durante la ejecución del software de aplicación.

En la forma de realización, el servidor puede referirse a un servidor en la nube que presenta un gran número de recursos informáticos, donde una máquina virtual de cada terminal que se comunica con el servidor en la nube puede estar instalada en el servidor en la nube. Una máquina virtual se refiere a un sistema completo que se simula mediante software, tiene funciones de sistema de hardware completas y se ejecuta en un entorno completamente aislado. Por lo general, una máquina virtual de múltiples terminales diferentes puede simularse en un servidor en la nube, y el terminal puede acceder, a través de una red, a la máquina virtual que es del terminal y que está instalada en el servidor en la nube.

Etapa 112: Analizar el evento agregado para obtener los múltiples eventos de entrada.

En la forma de realización, el servidor puede obtener en orden datos transmitidos de un evento de entrada actual a partir del evento agregado, donde los datos transmitidos incluyen información de mapa de bits de cada elemento de información de atributo del evento de entrada actual e información de atributo, que es diferente de la de un evento de entrada anterior, del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente del evento de entrada anterior; obtener, según la indicación de la información de mapa de bits, información de atributo, que es la misma que la del evento de entrada anterior, del evento de entrada actual a partir de la información de atributo del evento de entrada anterior; y combinar la información de atributo idéntica obtenida y la información de atributo diferente incluida en los datos transmitidos para formar el evento de entrada actual.

Etapa 113: Procesar los múltiples eventos de entrada usando una máquina virtual que es del terminal y que está instalada en el servidor para obtener datos de usuario generados durante la ejecución del software de aplicación.

En la forma de realización, cuando se recibe una notificación que indica que se ha producido un cambio en un estado de sistema de la máquina virtual que es el determina y que está instalada en el servidor, el servidor determina si la información de atributo de los múltiples eventos de entrada coincide con el estado de sistema e inserta un evento de entrada correspondiente a la información de atributo que coincide con el estado de sistema en un sistema de la máquina virtual, de manera que el sistema de la máquina virtual invoca una función de procesamiento para procesar el evento de entrada con el fin de obtener los datos de usuario generados durante la ejecución del software de aplicación.

Puede observarse que en la forma de realización, un terminal no necesita transmitir datos de usuario a un servidor, sino que transmite eventos de entrada al servidor de una manera basada en eventos, de modo que un servidor inicia la ejecución de los eventos de entrada usando una máquina virtual del terminal con el fin de obtener datos de usuario idénticos a los de un terminal que ejecuta software de aplicación, garantizándose así que el servidor pueda realizar una copia de seguridad completa de los datos de usuario y mejorándose la precisión de la comprobación de

seguridad en los datos de usuario. Además, debido a que el tamaño de un evento de entrada agregado transmitido por el terminal es relativamente pequeño, en comparación con el tamaño para transmitir directamente los datos de usuario, se utilizan menos recursos de transmisión de red y el rendimiento del propio terminal no se ve afectado, lo que garantiza la normal ejecución de otros servicios en el terminal.

5 Se hace referencia a la FIG. 2, que es un diagrama de flujo de otra forma de realización de un procedimiento para sincronizar información espejo de un terminal según la presente invención. En la forma de realización, un proceso de sincronización se describe por medio de una interacción entre un terminal y un servidor.

10 Etapa 201: Un terminal obtiene múltiples eventos de entrada durante la ejecución de software de aplicación.

Un software de aplicación diferente en el terminal puede iniciar un evento de entrada de un tipo diferente durante el tiempo de ejecución. Estos eventos de entrada pueden incluir un evento de entrada táctil para llevar a cabo una operación en una pantalla, un evento de entrada de posicionamiento generado por un dispositivo GPS, un evento de entrada de detección generado por un sensor, etc. La información de atributo incluida en un evento de entrada es diferente dependiendo de un tipo diferente del evento de entrada. Por ejemplo, para un evento de entrada táctil, la información de atributo incluida en el evento de entrada táctil puede incluir coordenadas de contacto del toque, el tipo de evento táctil, el tiempo en que se produce, etc. Para un evento de entrada de posicionamiento, la información de atributo incluida en el evento de entrada de posicionamiento puede incluir información de posición, el tiempo en que se produce, etc.

Etapa 202: Insertar los eventos de entrada obtenidos en una cola de eventos según una secuencia de generación de eventos.

25 En la forma de realización, en un proceso de ejecución del software de aplicación, cada vez que se genera un evento de entrada, el evento de entrada puede insertarse en la cola de eventos, donde la cola de eventos puede almacenarse en la memoria del terminal, y los eventos de entrada de la cola de eventos se ordenan de la manera 'primero en entrar primero en salir'. Por consiguiente, el terminal puede obtener en orden el evento de entrada a partir de la cola de eventos, y puede procesar el evento de entrada invocando una función de procesamiento que corresponde al evento de entrada, con el fin de ejecutar el software de aplicación en el terminal.

Etapa 203: Agregar los eventos de entrada en la cola de eventos según un periodo de tiempo fijado para obtener un evento agregado.

35 El terminal puede escribir los eventos de entrada almacenados en la cola de eventos de la memoria en un disco por medio de un procesamiento asíncrono. Además, en la forma de realización, el terminal puede agregar los múltiples eventos de entrada en la cola de eventos según un periodo de tiempo fijado para obtener el evento agregado. Debe observarse que, además de agregar los eventos de entrada según el periodo de tiempo, también puede fijarse un umbral de evento, y cuando el número de eventos de entrada en la cola de eventos alcanza el umbral de evento, los eventos de entrada se agregan. Esta forma de realización de la presente invención no está limitada a esto.

En la forma de realización, puesto que los múltiples eventos de entrada de la cola de eventos están ordenados de la manera 'primero en entrar, primero en salir' y múltiples eventos de entrada consecutivos son generalmente eventos iniciados para ejecutar el mismo software de aplicación, dos eventos de entrada adyacentes tienen generalmente la misma información de atributo. Por lo tanto, cuando se transmite un evento de entrada subsiguiente, solo hay que transmitir información de atributo que sea diferente de la de un evento de entrada anterior. Según esto, el terminal puede obtener, cuando agrega los múltiples eventos de entrada en la cola de eventos, un evento de entrada actual según una secuencia de generación de los múltiples eventos de entrada; generar información de mapa de bits para cada elemento de información de atributo del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente de una información de atributo anterior del evento de entrada actual; determinar la información de mapa de bits del evento de entrada actual e información de atributo, que es diferente de la del evento de entrada anterior, como datos que van a transmitirse del evento de entrada actual; y después de determinar los datos a transmitir de cada evento de entrada de los múltiples eventos de entrada, combinar los datos a transmitir de los múltiples eventos de entrada para formar el evento agregado de los múltiples eventos de entrada.

Etapa 204: Comprimir el evento agregado para obtener un evento agregado comprimido.

60 En la forma de realización, el evento agregado puede comprimirse para reducir adicionalmente el tamaño de un flujo de datos del evento agregado que va a transmitirse. Un modo de compresión puede ser la compresión ZIP, la compresión RAR o similar. Esta forma de realización de la presente invención no está limitada a esto.

Etapa 205: Transmitir el evento agregado comprimido a un servidor.

65 Etapa 206: El servidor descomprime el evento agregado comprimido para obtener el evento agregado.

En esta etapa, la manera en que el servidor descomprime el evento agregado comprimido tras recibir el evento agregado comprimido transmitido por el terminal es compatible con la manera en que el terminal comprime el evento agregado en la etapa 204. Una vez que se ha descomprimido el evento agregado comprimido, puede obtenerse el evento agregado.

5 Etapa 207: Analizar el evento agregado para obtener los múltiples eventos de entrada.

10 Con referencia a un proceso en que el terminal agrega los eventos de entrada para obtener el evento agregado en la etapa 203, en esta etapa el servidor puede obtener en orden datos transmitidos de un evento de entrada actual a partir del evento agregado, donde los datos transmitidos incluyen información de mapa de bits de cada elemento de información de atributo del evento de entrada actual e información de atributo, que es diferente de la del evento de entrada anterior, del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente de la información de atributo anterior; obtener, según la indicación de la información de mapa de bits, información de atributo, que es la misma que la del evento de entrada anterior, del evento de entrada actual a partir de la información de atributo del evento de entrada anterior; y combinar la información de atributo idéntica obtenida y la información de atributo diferente incluida en los datos transmitidos para formar el evento de entrada actual.

20 Etapa 208: Procesar los múltiples eventos de entrada usando una máquina virtual que es del terminal y que está instalada en el servidor para obtener datos de usuario generados durante la ejecución del software de aplicación.

25 En esta etapa, cuando el servidor recibe una notificación que indica que se ha producido un cambio en un estado de sistema de la máquina virtual que es el determina y que está instalada en el servidor, el servidor determina si la información de atributo de los múltiples eventos de entrada coincide con el estado de sistema, e inserta un evento de entrada correspondiente a la información de atributo que coincide con el estado de sistema en un sistema de la máquina virtual, de manera que el sistema de la máquina virtual invoca una función de procesamiento para procesar el evento de entrada, donde el procesamiento es el mismo que el procesamiento del evento de entrada en el terminal al invocar la función de procesamiento. Es decir, un proceso que ejecuta el software de aplicación en la máquina virtual del terminal en el servidor es el mismo que el del terminal. Por consiguiente, los datos de usuario generados en la máquina virtual del terminal son idénticos a los datos de usuario generados por el terminal, consiguiéndose así el objetivo de sincronizar información espejo de un terminal en el servidor.

35 La información de atributo del evento de entrada incluye generalmente múltiples elementos. La información de atributo que coincide con el estado de sistema de la máquina virtual del terminal puede ser al menos un elemento de los múltiples elementos de la información de atributo. Por ejemplo, para un evento de entrada táctil, puede determinarse si el tiempo de aparición incluido en su información de atributo es el mismo que el tiempo actual de un sistema de la máquina virtual del terminal, y si el tiempo de aparición incluido en su información de atributo es el mismo que el tiempo actual del sistema de la máquina virtual del terminal, el evento de entrada táctil se inserta en el sistema.

45 Puede observarse que en la forma de realización, un terminal no necesita transmitir datos de usuario a un servidor, sino que transmite eventos de entrada al servidor de una manera basada en eventos, de modo que un servidor inicia la ejecución de los eventos de entrada usando una máquina virtual del terminal con el fin de obtener datos de usuario idénticos a los de un terminal que ejecuta software de aplicación, garantizándose así que el servidor pueda realizar una copia de seguridad completa de los datos de usuario y mejorándose la precisión de la comprobación de seguridad en los datos de usuario. Además, debido a que el tamaño de un evento de entrada agregado transmitido por el terminal es relativamente pequeño, en comparación con el tamaño para transmitir directamente los datos de usuario, se utilizan menos recursos de transmisión en red y el rendimiento del propio terminal no se ve afectado, lo que garantiza la normal ejecución de otros servicios en el terminal.

Se hace referencia a la FIG. 3A, que es un diagrama esquemático de una arquitectura de red según una forma de realización de la presente invención.

55 La arquitectura de red mostrada en la FIG. 3A incluye un terminal inteligente usado por un usuario y un servidor en la nube, donde el terminal inteligente y el servidor en la nube se comunican a través de una red. El terminal inteligente incluye un módulo de puesta en cola de eventos, un módulo de procesamiento de eventos y un módulo de servicio de registro. El servidor en la nube incluye un módulo de planificación de eventos, un módulo de inserción de eventos y un módulo de procesamiento de eventos.

60 Supóngase que el terminal inteligente mostrado en la FIG. 3A tiene una pantalla táctil. A continuación se describe un proceso de sincronización de información espejo de un terminal desde el terminal inteligente al servidor en la nube usando un ejemplo en el que un usuario hace clic sobre un icono de software de aplicación en la pantalla táctil.

65 En un terminal inteligente:

El usuario hace clic, cuando trata de ejecutar el software de aplicación, sobre el icono del software de aplicación en la pantalla táctil. En este caso, el módulo de puesta en cola de eventos puede abstraer en un evento de entrada la acción de usuario de hacer clic sobre el icono usando un controlador *kernel*, y puede insertar el evento de entrada en una cola de eventos dentro del terminal inteligente. Después, cuando el usuario lleva a cabo una operación en el software de aplicación iniciado, un evento de entrada diferente se activa de manera correspondiente en un proceso de funcionamiento, y estos eventos de entrada se insertan en la cola de eventos dentro del terminal inteligente, donde los eventos de entrada de la cola de eventos se ordenan de la manera 'primero en entrar, primero en salir'.

5 El módulo de procesamiento de eventos puede obtener en orden un evento de entrada de la cola de eventos usando un hilo de procesamiento de eventos, y puede procesar el evento de entrada invocando una función de procesamiento correspondiente al evento de entrada, con el fin de ejecutar el software de aplicación en el terminal inteligente. Además, el módulo de procesamiento de eventos también puede enviar al módulo de servicio de registro los eventos de entrada de la cola de eventos por medio de un procesamiento asíncrono.

10 15 Tras obtener los múltiples eventos de entrada, el módulo de servicio de registro puede agregar y comprimir estos eventos de entrada para obtener un evento agregado comprimido, y transmitir el evento agregado comprimido al servidor en la nube a través de la red. En la forma de realización, el módulo de servicio de registro puede agregar y comprimir los eventos de entrada obtenidos según un periodo de tiempo fijado. Por ejemplo, puede establecerse que los eventos de entrada obtenidos se agreguen y compriman en un intervalo de 5 s.

20 Cuando el módulo de servicio de registro agrega los eventos de entrada, puesto que los múltiples eventos de entrada obtenidos por el módulo de servicio de registro están ordenados de la manera 'primero en entrar, primero en salir' en la cola de eventos, y múltiples eventos de entrada consecutivos son generalmente eventos activados para ejecutar el mismo software de aplicación, dos eventos de entrada adyacentes tienen generalmente la misma información de atributo. Por lo tanto, cuando se transmite un evento de entrada subsiguiente, se permite que solo haya que transmitir información de atributo que sea diferente de la de un evento de entrada anterior del mismo, reduciéndose así el tamaño de todo el evento agregado comprimido y reduciéndose el número de recursos de red utilizados para transmitir el evento agregado comprimido. Como se muestra en la FIG. 3B, supóngase que el módulo de servicio de registro transmite tres eventos de entrada esta vez, que son el evento 1, el evento 2 y el evento 3, donde cada evento de entrada está formado por cuatro elementos de información de atributo. Una cabecera de cada evento de entrada tiene información de mapa de bits que corresponde, respectivamente, a cada elemento de información de atributo. Un valor de bit de cada elemento de información de mapa de bits se usa para indicar si la información de atributo correspondiente a la información de mapa de bits es la misma que la información de atributo de un evento de entrada anterior. Por ejemplo, si la información de atributo correspondiente a la información de mapa de bits es la misma que la información de atributo de un evento de entrada anterior, el valor de bit de la información de mapa de bits es 1; y si la información de atributo correspondiente a la información de mapa de bits es diferente de la información de atributo de un evento de entrada anterior, el valor de bit de la información de mapa de bits es 0. Con referencia a la FIG. 3B, puede observarse que cuatro elementos de información de atributo del evento 1 son, respectivamente, A, B, C y D. Supóngase que ninguno de los cuatro atributos del evento 1 coincide con los de un evento anterior. Por consiguiente, todos los valores de bit de la información de mapa de bits de los cuatro elementos de información de atributo del evento 1 valen 0. Para el evento 2, un segundo elemento de información de atributo y un cuarto elemento de información de atributo del mismo son idénticos a los del evento 1, y un primer elemento de información de atributo y un tercer elemento de información de atributo son diferentes de los del evento 1, que son, respectivamente, E y F. Por lo tanto, cuatro valores de bit de la información de mapa de bits del evento 2 son 0, 1, 0 y 1. Para el evento 3, un primer elemento de información de atributo, un segundo elemento de información de atributo y un cuarto elemento de información de atributo del mismo son idénticos a los del evento 2, y un tercer elemento de información de atributo es diferente del atributo del evento 2, que es G. Por lo tanto, cuatro valores de bit de la información de mapa de bits del evento 3 son 1, 1, 0 y 1. A partir de la descripción anterior puede observarse que cuando se transmiten los tres eventos de entrada mostrados en la FIG. 3B, aunque los tres eventos de entrada tienen un total de 12 elementos de información de atributo, al fijarse la información de mapa de bits para cada evento solo es necesario transmitir realmente siete elementos de información de atributo. Puesto que la información de mapa de bits ocupa un volumen de bits más pequeño, se reduce el tamaño de un flujo de bits global para transmitir los tres eventos de entrada.

55 Además, para un flujo de bits de eventos de entrada agregados que va a transmitirse, el módulo de servicio de registro puede comprimir el flujo de datos para reducir adicionalmente el tamaño del flujo de datos. Un modo de compresión puede ser la compresión ZIP, la compresión RAR o similar. Esta forma de realización de la presente invención no está limitada a esto.

60 En un servidor en la nube:

El módulo de planificación de eventos descomprime, tras recibir el flujo de datos de los eventos de entrada agregados y comprimidos transmitidos por el módulo de servicio de registro del terminal inteligente, el flujo de datos para obtener los eventos de entrada agregados, y después analiza los eventos de entrada agregados para obtener los múltiples eventos de entrada.

La manera en que el módulo de planificación de eventos descomprime el flujo de datos es compatible con la manera en que el módulo de servicio de registro comprime el flujo de datos. Cuando el módulo de planificación de eventos analiza los eventos de entrada agregados descomprimidos, con referencia a la FIG. 3B, puede observarse que después de obtenerse la información de atributo y la información de mapa de bits de los tres eventos de entrada, excepto del evento 1, cada uno de los siguientes eventos de entrada puede restaurarse según la información de mapa de bits del evento de entrada y la información de atributo del evento de entrada anterior del evento de entrada. Por ejemplo, para el evento 2, puede observarse, según los valores de bit "0101" de su información de mapa de bits, que el segundo elemento de información de atributo y el cuarto elemento de información de atributo son idénticos a los del evento 1; por lo tanto, los dos elementos de información de atributo se obtienen a partir del evento 1. En lo que respecta al evento 3, puede observarse, según los valores de bit "1101" de su información de mapa de bits, que el primer elemento de información de atributo, el segundo elemento de información de atributo y el cuarto elemento de información de atributo del mismo son idénticos a los del evento 2; por lo tanto, los tres elementos de información de atributo se obtienen a partir del evento 2.

El módulo de inserción de eventos determina si la información de atributo, que se obtiene analizando el módulo de planificación de eventos, de cada evento de entrada coincide con un estado de sistema de una máquina virtual del terminal inteligente; y si la información de atributo de cada evento de entrada coincide con el estado de sistema de la máquina virtual del terminal inteligente, inserta el evento de entrada en el sistema. La máquina virtual del terminal inteligente puede activar en un modo de notificación, cuando se produce un cambio en el estado de sistema de la misma, el módulo de inserción de eventos para hacer corresponder la información de atributo del evento de entrada con el estado de sistema.

En esta forma de realización, la información de atributo del evento de entrada incluye generalmente múltiples elementos. La información de atributo que coincide con el estado de sistema de la máquina virtual del terminal inteligente puede ser al menos un elemento de los múltiples elementos de la información de atributo. Por ejemplo, para un evento de entrada táctil, puede determinarse si el tiempo de aparición incluido en su información de atributo es el mismo que el tiempo de un sistema actual de la máquina virtual del terminal inteligente, y si el tiempo de aparición incluido en su información de atributo es el mismo que el tiempo del sistema actual de la máquina virtual del terminal inteligente, el evento de entrada táctil se inserta en el sistema.

El módulo de procesamiento de eventos procesa, después de que el módulo de inserción de eventos inserte el evento de entrada en el sistema, el evento de entrada invocando una función de procesamiento correspondiente de la máquina virtual del terminal inteligente, donde el procesamiento es el mismo que el procesamiento del evento de entrada del módulo de procesamiento de eventos al invocar una función de procesamiento en el terminal inteligente. Es decir, los procesos que ejecutan el software de aplicación en la máquina virtual del terminal inteligente en el servidor en la nube y en el terminal inteligente son idénticos. Por consiguiente, los datos de usuario generados en la máquina virtual del terminal inteligente son también idénticos a los datos de usuario generados en el terminal inteligente, consiguiéndose así el objetivo de sincronizar información espejo de un terminal en el servidor en la nube.

Usando la forma de realización anterior, el terminal inteligente necesita transmitir solamente los eventos de entrada al servidor en la nube para realizar una copia de seguridad de los datos de usuario del terminal inteligente en el servidor en la nube, de modo que cuando el terminal inteligente se extravía, todos los datos de usuario del terminal inteligente, incluidos los datos de estado y los datos de archivo antes de su pérdida, pueden restaurarse desde el servidor en la nube. Como alternativa, usando la gran cantidad de recursos informáticos del servidor en la nube, se lleva a cabo una exploración de seguridad de los datos de usuario en el servidor en la nube, con el fin de obtener un resultado de detección que indica si los datos de usuario del terminal inteligente están protegidos.

De manera correspondiente a las formas de realización del procedimiento para sincronizar información espejo de un terminal según la presente invención, la presente invención proporciona además formas de realización de un aparato, un terminal y un servidor para sincronizar información espejo de un terminal para ejecutar el procedimiento.

Se hace referencia a la FIG. 4, que es un diagrama de bloques de una forma de realización de un aparato para sincronizar información espejo de un terminal según la presente invención. El aparato puede estar dispuesto en un terminal.

El aparato incluye una unidad de obtención 410, una unidad de agregación 420 y una unidad de transmisión 430.

La unidad de obtención 410 está configurada para obtener múltiples eventos de entrada durante la ejecución del software de aplicación.

La unidad de agregación 420 está configurada para agregar los múltiples eventos de entrada obtenidos por la unidad de obtención 410 con el fin de obtener un evento agregado.

La unidad de transmisión 430 está configurada para transmitir a un servidor el evento agregado obtenido por la unidad de agregación 420, de modo que tras analizar el evento agregado para obtener los múltiples eventos de entrada, el servidor procesa los múltiples eventos de entrada usando una máquina virtual que es del terminal y está

instalada en el servidor, con el fin de obtener datos de usuario generados durante la ejecución del software de aplicación.

La unidad de agregación 420 puede incluir (aunque no se muestra en la FIG. 4):

- 5 una subunidad de obtención de eventos, configurada para obtener un evento de entrada actual según una secuencia de generación de los múltiples eventos de entrada;
- 10 una subunidad de generación de información, configurada para generar información de mapa de bits para cada elemento de información de atributo del evento de entrada actual obtenido por la subunidad de obtención de eventos, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente de un evento de entrada anterior del evento de entrada actual;
- 15 una subunidad de determinación de datos, configurada para determinar la información de mapa de bits, que se genera mediante la subunidad de generación de información, del evento de entrada actual e información de atributo, que es diferente de la del evento de entrada anterior, como datos que van a transmitirse del evento de entrada actual; y
- 20 una subunidad de agregación de eventos, configurada para combinar, después de que la subunidad de determinación de datos determine los datos que van a transmitirse de cada evento de entrada de los múltiples eventos de entrada, los datos que van a transmitirse de los múltiples eventos de entrada para formar el evento agregado de los múltiples eventos de entrada.
- 25 Se hace referencia a la FIG. 5, que es un diagrama de bloques de otra forma de realización de un aparato para sincronizar información espejo de un terminal según la presente invención. El aparato puede estar dispuesto en un terminal.
- 30 El aparato incluye una unidad de obtención 510, una unidad de inserción 520, una unidad de agregación 530, una unidad de compresión 540 y una unidad de transmisión 550.
- La unidad de obtención 510 está configurada para obtener múltiples eventos de entrada durante la ejecución del software de aplicación.
- 35 La unidad de inserción 520 está configurada para insertar los eventos de entrada obtenidos por la unidad de obtención 510 en una cola de eventos según una secuencia de generación de eventos. La unidad de agregación 530 está configurada para agregar los eventos de entrada en la cola de eventos según un periodo de tiempo fijado para obtener el evento agregado.
- 40 La unidad de compresión 540 está configurada para comprimir el evento agregado obtenido por la unidad de agregación 540, con el fin de obtener un evento agregado comprimido.
- 45 La unidad de transmisión 550 está configurada específicamente para transmitir a un servidor en la nube el evento agregado comprimido obtenido por la unidad de compresión 540, de modo que tras analizar el evento agregado para obtener los múltiples eventos de entrada, el servidor procesa los múltiples eventos de entrada usando una máquina virtual que es del terminal y está instalada en el servidor, con el fin de obtener datos de usuario generados durante la ejecución del software de aplicación.
- La unidad de agregación 530 puede incluir (aunque no se muestra en la FIG. 5):
- 50 una subunidad de obtención de eventos, configurada para obtener un evento de entrada actual según una secuencia de generación de los múltiples eventos de entrada;
- 55 una subunidad de generación de información, configurada para generar información de mapa de bits para cada elemento de información de atributo del evento de entrada actual obtenido por la subunidad de obtención de eventos, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente de un evento de entrada anterior del evento de entrada actual;
- 60 una subunidad de determinación de datos, configurada para determinar la información de mapa de bits, que se genera mediante la subunidad de generación de información, del evento de entrada actual e información de atributo, que es diferente de la del evento de entrada anterior, como datos que van a transmitirse del evento de entrada actual; y
- 65 una subunidad de agregación de eventos, configurada para combinar, después de que la subunidad de determinación de datos determine los datos que van a transmitirse de cada evento de entrada de los

múltiples eventos de entrada, los datos que van a transmitirse de los múltiples eventos de entrada para formar el evento agregado de los múltiples eventos de entrada.

5 Se hace referencia a la FIG. 6, que es un diagrama de bloques de una forma de realización de un terminal según la presente invención.

El terminal incluye un procesador 610 y una interfaz de red 620.

10 El procesador 610 está configurado para obtener múltiples eventos de entrada durante la ejecución de software de aplicación, y para agregar los múltiples eventos de entrada para obtener un evento agregado.

15 La interfaz de red 620 está configurada para transmitir el evento agregado a un servidor, de modo que tras analizar el evento agregado para obtener los múltiples eventos de entrada, el servidor procesa los múltiples eventos de entrada usando una máquina virtual que es del terminal y que está instalada en el servidor, con el fin de obtener datos de usuario generados durante la ejecución del software de aplicación.

En una implementación opcional:

20 el procesador 610 puede estar configurado además para insertar los eventos de entrada obtenidos en una cola de eventos según una secuencia de generación de eventos; y

el procesador 610 puede estar configurado específicamente para agregar los eventos de entrada en la cola de eventos según un periodo de tiempo fijado para obtener el evento agregado.

25 En otra implementación opcional:

30 el procesador 610 puede estar configurado específicamente para: obtener un evento de entrada actual según una secuencia de generación de los múltiples eventos de entrada; generar información de mapa de bits para cada elemento de información de atributo del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente de un evento de entrada anterior del evento de entrada actual; determinar la información de mapa de bits e información de atributo, que es diferente de la del evento de entrada anterior, del evento de entrada actual como datos que van a transmitirse del evento de entrada actual; y después de determinar los datos a transmitir de cada evento de entrada de los múltiples eventos de entrada, combinar los datos a transmitir de los múltiples eventos de entrada para formar el evento agregado de los múltiples eventos de entrada.

En otra implementación opcional:

40 el procesador 610 puede estar configurado además para comprimir el evento agregado para obtener un evento agregado comprimido; y

45 la interfaz de red 620 puede estar configurada específicamente para transmitir el evento agregado comprimido a un servidor en la nube.

Se hace referencia a la FIG. 7, que es un diagrama de bloques de otra forma de realización de un aparato para sincronizar información espejo de un terminal según la presente invención. El aparato puede estar dispuesto en un servidor.

50 El aparato incluye una unidad de recepción 710, una unidad de análisis 720 y una unidad de procesamiento 730.

55 La unidad de recepción 710 está configurada para recibir un evento agregado transmitido por un terminal, donde el evento agregado es un evento agregado obtenido por el terminal agregando múltiples eventos de entrada después de obtener los múltiples eventos de entrada durante la ejecución del software de aplicación.

La unidad de análisis 720 está configurada para analizar el evento agregado recibido por la unidad de recepción 710, con el fin de obtener los múltiples eventos de entrada.

60 La unidad de procesamiento 730 está configurada para procesar, usando una máquina virtual que es del terminal y que está instalada en el servidor, los múltiples eventos de entrada obtenidos por la unidad de análisis 720, con el fin de obtener datos de usuario generados durante la ejecución del software de aplicación.

La unidad de análisis 720 puede incluir (aunque no se muestra en la FIG. 7):

65 una subunidad de obtención de datos, configurada para obtener en orden datos transmitidos de un evento de entrada actual del evento agregado, donde los datos transmitidos incluyen información de mapa de bits de

cada elemento de información de atributo del evento de entrada actual e información de atributo, que es diferente de la de un evento de entrada anterior, del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente del evento de entrada anterior;

5 una subunidad de obtención de atributos, configurada para obtener, según la indicación de la información de mapa de bits obtenida por la subunidad de obtención de datos, información de atributo, que es la misma que la del evento de entrada anterior, del evento de entrada actual a partir de la información de atributo del evento de entrada anterior; y

10 una subunidad de generación de eventos, configurada para combinar la misma información de atributo obtenida por la subunidad de obtención de atributos y la diferente información de atributo incluida en los datos transmitidos obtenidos por la subunidad de obtención de datos para formar el evento de entrada actual.

15 La unidad de procesamiento 730 puede incluir (aunque no se muestra en la FIG. 7):

una subunidad de correspondencia de estados, configurada para: cuando se recibe una notificación que indica que se ha producido un cambio en un estado de sistema de la máquina virtual que es del terminal y que está instalada en el servidor, determinar si la información de atributo de los múltiples eventos de entrada coincide con el estado de sistema; y

20 una subunidad de procesamiento de eventos, configurada para: según un resultado de correspondencia de la subunidad de correspondencia de estados, insertar un evento de entrada correspondiente a la información de atributo que coincide con el estado de sistema en un sistema de la máquina virtual, de modo que el sistema invoque una función de procesamiento para procesar el evento de entrada.

Se hace referencia a la FIG. 8, que es un diagrama de bloques de otra forma de realización de un aparato para sincronizar información espejo de un terminal según la presente invención. El aparato puede estar dispuesto en un servidor.

30 El aparato incluye una unidad de recepción 810, una unidad de descompresión 820, una unidad de análisis 830 y una unidad de procesamiento 840.

35 La unidad de recepción 810 está configurada para recibir un evento agregado comprimido transmitido por un terminal, donde el evento agregado comprimido es un evento obtenido por el terminal agregando múltiples eventos de entrada después de obtener los múltiples eventos de entrada durante la ejecución del software de aplicación, y después comprimiendo el evento agregado.

40 La unidad de descompresión 820 está configurada para descomprimir el evento agregado comprimido recibido por la unidad de recepción 810, con el fin de obtener el evento agregado.

La unidad de análisis 830 está configurada para analizar el evento agregado obtenido por la unidad de descompresión 820, con el fin de obtener los múltiples eventos de entrada.

45 La unidad de procesamiento 840 está configurada para procesar, usando una máquina virtual que es del terminal y que está instalada en el servidor, los múltiples eventos de entrada obtenidos por la unidad de análisis 830, con el fin de obtener datos de usuario generados durante la ejecución del software de aplicación.

50 La unidad de análisis 830 puede incluir (aunque no se muestra en la FIG. 8):

una subunidad de obtención de datos, configurada para obtener en orden los datos transmitidos de un evento de entrada actual del evento agregado, donde los datos transmitidos incluyen información de mapa de bits de cada elemento de información de atributo del evento de entrada actual e información de atributo, que es diferente de la de un evento de entrada anterior, del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente del evento de entrada anterior;

55 una subunidad de obtención de atributos, configurada para obtener, según la indicación de la información de mapa de bits obtenida por la subunidad de obtención de datos, información de atributo, que es la misma que la del evento de entrada anterior, del evento de entrada actual a partir de la información de atributo del evento de entrada anterior; y

60 una subunidad de generación de eventos, configurada para combinar la misma información de atributo obtenida por la subunidad de obtención de atributos y la diferente información de atributo incluida en los datos transmitidos obtenidos por la subunidad de obtención de datos para formar el evento de entrada actual.

65

La unidad de procesamiento 840 puede incluir (aunque no se muestra en la FIG. 8):

5 una subunidad de correspondencia de estados, configurada para: cuando se recibe una notificación que indica que se ha producido un cambio en un estado de sistema de la máquina virtual que es del terminal y que está instalada en el servidor, determinar si la información de atributo de los múltiples eventos de entrada coincide con el estado de sistema; y

10 una subunidad de procesamiento de eventos, configurada para: según un resultado de correspondencia de la subunidad de correspondencia de estados, insertar un evento de entrada correspondiente a la información de atributo que coincide con el estado de sistema en un sistema de la máquina virtual, de modo que el sistema invoque una función de procesamiento para procesar el evento de entrada.

15 Se hace referencia a la FIG. 9, que es un diagrama de bloques de una forma de realización de un servidor según la presente invención.

El servidor incluye una interfaz de terminal 910 y un procesador 920.

20 La interfaz de terminal 910 está configurada para recibir un evento agregado transmitido por un terminal, donde el evento agregado es un evento agregado obtenido por el terminal agregando múltiples eventos de entrada después de obtener los múltiples eventos de entrada durante la ejecución del software de aplicación.

25 El procesador 920 está configurado para analizar el evento agregado para obtener los múltiples eventos de entrada, y procesa los múltiples eventos de entrada usando una máquina virtual que es del terminal y que está instalada en el servidor, con el fin de obtener datos de usuario generados durante la ejecución del software de aplicación.

En una implementación opcional:

30 La interfaz de terminal 910 puede estar configurada específicamente para recibir un evento agregado comprimido transmitido por el terminal, donde el evento agregado comprimido es un evento obtenido por el terminal tras comprimir el evento agregado.

El procesador 920 puede estar configurado además para descomprimir el evento agregado comprimido para obtener el evento agregado.

35 En otra implementación opcional:

40 El procesador 920 puede estar configurado específicamente para: obtener en orden datos transmitidos de un evento de entrada actual a partir del evento agregado, donde los datos transmitidos incluyen información de mapa de bits de cada elemento de información de atributo del evento de entrada actual e información de atributo, que es diferente de la de un evento de entrada anterior, del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente del evento de entrada anterior; obtener, según la indicación de la información de mapa de bits, información de atributo, que es la misma que la del evento de entrada anterior, del evento de entrada actual a partir de la información de atributo del evento de entrada anterior; y combinar la información de atributo idéntica obtenida y la información de atributo diferente incluida en los datos transmitidos para formar el evento de entrada actual.

En otra implementación opcional:

50 El procesador 920 puede estar configurado específicamente para: cuando se recibe una notificación que indica que se ha producido un cambio en un estado de sistema de la máquina virtual que es el determina y que está instalada en el servidor, determinar si la información de atributo de los múltiples eventos de entrada coincide con el estado de sistema; e insertar un evento de entrada correspondiente a la información de atributo que coincide con el estado de sistema en un sistema de la máquina virtual, de manera que el sistema invoque una función de procesamiento para procesar el evento de entrada.

60 Puede observarse que en la forma de realización, un terminal obtiene múltiples eventos de entrada durante la ejecución de software de aplicación, agrega los múltiples eventos de entrada para obtener un evento agregado y transmite el evento agregado a un servidor; tras analizar el evento agregado para obtener los múltiples eventos de entrada, el servidor procesa los múltiples eventos de entrada usando una máquina virtual que es del terminal y que está instalada en el servidor, con el fin de obtener datos de usuario generados durante la ejecución del software de aplicación. En las formas de realización de la presente invención, el terminal no necesita transmitir los datos de usuario al servidor, sino que transmite los eventos de entrada al servidor de una manera basada en eventos, de modo que un servidor inicia la ejecución de los eventos de entrada usando la máquina virtual del terminal con el fin de obtener datos de usuario idénticos a los de un terminal que ejecuta el software de aplicación, garantizándose así
65 que el servidor pueda realizar una copia de seguridad completa de los datos de usuario y mejorándose la precisión

de la comprobación de seguridad en los datos de usuario. Además, debido a que el tamaño de un evento de entrada agregado transmitido por el terminal es relativamente pequeño, en comparación con el tamaño para transmitir directamente los datos de usuario, se utilizan menos recursos de transmisión en red y el rendimiento del propio terminal no se ve afectado, lo que garantiza la normal ejecución de otros servicios en el terminal.

5 Un experto en la técnica puede entender claramente que la tecnología de las formas de realización de la presente invención puede implementarse usando software y una plataforma de hardware general necesaria. En base a esto, las soluciones técnicas de las formas de realización de la presente invención, o de la parte relativa a la técnica anterior, pueden implementarse en forma de un producto de software. El producto de software informático está
10 almacenado en un medio de almacenamiento, tal como una RAM/ROM, un disco magnético o un disco óptico, e incluye varias instrucciones para hacer que un dispositivo informático (que puede ser un ordenador personal, un servidor, un dispositivo de red o similar) ejecute los procedimientos descritos en las formas de realización de la presente invención o en algunas partes de las formas de realización de la presente invención.

15 Cada forma de realización de la memoria descriptiva se ha descrito de manera progresiva. Partes idénticas o similares de las formas de realización son simplemente referencias entre las mismas. Cada forma de realización resalta lo que es diferente de las otras formas de realización. En particular, para la forma de realización del sistema, puesto que es básicamente similar a la forma de realización del procedimiento, la forma de realización del sistema se describe de manera sencilla, y la parte relevante puede obtenerse con referencia a la parte de la descripción de la
20 forma de realización del procedimiento. No debe considerarse que las anteriores formas de realización de la presente invención limitan el alcance de protección de la presente invención. Cualquier modificación, equivalencia y mejora se realiza dentro del alcance de la presente invención, definida en las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un procedimiento para sincronizar información de espejo de un terminal, que comprende:

- 5 obtener (101, 111, 201), mediante un terminal, múltiples eventos de entrada durante la ejecución de software de aplicación;
- agregar (102, 111, 203), mediante el terminal, los múltiples eventos de entrada para obtener un evento agregado;
- 10 transmitir (103, 111, 205), mediante el terminal, el evento agregado a un servidor;
- analizar (103, 112, 207), mediante el servidor, el evento agregado para obtener los múltiples eventos de entrada; y
- 15 procesar (103, 113, 208), mediante el servidor, los múltiples eventos de entrada mediante software de aplicación ejecutado en una máquina virtual del terminal en el servidor para obtener datos de usuario generados durante la ejecución del software de aplicación.

20 2. El procedimiento según la reivindicación 1, en el que antes de agregar, mediante el terminal, los múltiples eventos de entrada, el procedimiento comprende además: insertar (202), mediante el terminal, los eventos de entrada obtenidos en una cola de eventos según una secuencia de generación de eventos; y, de manera correspondiente, la agregación, mediante el terminal, de los múltiples eventos de entrada para obtener un evento agregado comprende:

- 25 agregar, mediante el terminal, los eventos de entrada en la cola de eventos según un periodo de tiempo fijado para obtener el evento agregado.

3. El procedimiento según la reivindicación 1 o 2, en el que agregar, mediante el terminal, los múltiples eventos de entrada para obtener un evento agregado comprende:

- 30 obtener, mediante el terminal, un evento de entrada actual según una secuencia de generación de los múltiples eventos de entrada;
- generar, mediante el terminal, información de mapa de bits para cada elemento de información de atributo del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente de un evento de entrada anterior del evento de entrada actual;
- 35 determinar, mediante el terminal, la información de mapa de bits y la información de atributo, que es diferente de la del evento de entrada anterior, del evento de entrada actual como datos que van a transmitirse del evento de entrada actual; y
- 40 combinar, mediante el terminal tras determinarse datos que van a transmitirse de cada evento de entrada de los múltiples eventos de entrada, los datos que van a transmitirse de los múltiples eventos de entrada para formar el evento agregado de los múltiples eventos de entrada.
- 45

4. El procedimiento según cualquiera de las reivindicaciones 1 a 3, en el que el evento agregado transmitido por un terminal al servidor es un evento agregado comprimido obtenido por el terminal después de comprimir (204) el evento agregado; y antes de analizar, mediante el servidor, el evento agregado, el procedimiento comprende además:

- descomprimir (206), mediante el servidor, el evento agregado comprimido para obtener el evento agregado.

5. El procedimiento según cualquiera de las reivindicaciones anteriores, en el que analizar, mediante el servidor, el evento agregado para obtener los múltiples eventos de entrada comprende:

- 60 obtener en orden, mediante el servidor, los datos transmitidos de un evento de entrada actual a partir del evento agregado, donde los datos transmitidos comprenden información de mapa de bits de cada elemento de información de atributo del evento de entrada actual e información de atributo, que es diferente de la de un evento de entrada anterior, del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente del evento de entrada anterior;
- 65 obtener, mediante el servidor según la indicación de la información de mapa de bits, información de atributo, que es la misma que la del evento de entrada anterior, del evento de entrada actual a partir de la información de atributo del evento de entrada anterior; y

combinar, mediante el servidor, la información de atributo idéntica obtenida y la información de atributo diferente comprendida en los datos transmitidos para formar el evento de entrada actual.

5 6. El procedimiento según cualquiera de las reivindicaciones anteriores, en el que procesar, mediante el servidor, los múltiples eventos de entrada comprende:

cuando el servidor recibe una notificación que indica que se ha producido un cambio en un estado de sistema de la máquina virtual, determinar si la información de atributo de los múltiples eventos de entrada coincide con el estado de sistema; e

10 insertar, mediante el servidor, un evento de entrada correspondiente a la información de atributo que coincide con el estado de sistema en un sistema de la máquina virtual, de modo que el sistema invoque una función de procesamiento para procesar el evento de entrada.

15 7. Un terminal, que comprende un procesador (610) y una interfaz de red (620), en el que:

20 el procesador (610) está configurado para obtener múltiples eventos de entrada durante la ejecución de software de aplicación, y para agregar los múltiples eventos de entrada para obtener un evento agregado; y la interfaz de red (620) está configurada para transmitir el evento agregado a un servidor, de modo que tras analizar el evento agregado para obtener los múltiples eventos de entrada, el servidor procesa los múltiples eventos de entrada mediante software de aplicación ejecutado en una máquina virtual del terminal en el servidor para obtener datos de usuario generados durante la ejecución del software de aplicación.

25 8. El terminal según la reivindicación 7, en el que:

el procesador (610) está configurado además para insertar los eventos de entrada obtenidos en una cola de eventos según una secuencia de generación de eventos; y

30 el procesador (610) está configurado específicamente para agregar los eventos de entrada en la cola de eventos según un periodo de tiempo fijado para obtener el evento agregado.

9. El terminal según la reivindicación 7 u 8, en el que:

35 el procesador (610) está configurado específicamente para: obtener un evento de entrada actual según una secuencia de generación de los múltiples eventos de entrada; generar información de mapa de bits para cada elemento de información de atributo del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente de un evento de entrada anterior del evento de entrada actual; determinar la información de mapa de bits e información de atributo, que es diferente de la del evento de entrada anterior, del evento de entrada actual como datos que van a transmitirse del evento de entrada actual; y después de determinar los datos a transmitir de cada evento de entrada de los múltiples eventos de entrada, combinar los datos a transmitir de los múltiples eventos de entrada para formar el evento agregado de los múltiples eventos de entrada.

45 10. Un servidor, que comprende una interfaz de terminal (910) y un procesador (920), en el que:

50 la interfaz de terminal (910) está configurada para recibir un evento agregado transmitido por un terminal, donde el evento agregado es un evento agregado obtenido por el terminal agregando múltiples eventos de entrada después de obtener los múltiples eventos de entrada durante la ejecución del software de aplicación; y

55 el procesador (920) está configurado para analizar el evento agregado para obtener los múltiples eventos de entrada, y procesar los múltiples eventos de entrada mediante software de aplicación ejecutado en una máquina virtual del terminal en el servidor para obtener datos de usuario generados durante la ejecución del software de aplicación.

11. El servidor según la reivindicación 10, en el que:

60 la interfaz de terminal (910) está configurada específicamente para recibir un evento agregado comprimido transmitido por el terminal, donde el evento agregado comprimido es un evento obtenido por el terminal tras comprimir el evento agregado; y

65 el procesador (920) está configurado además para descomprimir el evento agregado comprimido para obtener el evento agregado.

12. El servidor según la reivindicación 10 u 11, en el que:

5 el procesador (920) está configurado específicamente para: obtener en orden datos transmitidos de un evento de entrada actual a partir del evento agregado, donde los datos transmitidos comprenden información de mapa de bits de cada elemento de información de atributo del evento de entrada actual e información de atributo, que es diferente de la de un evento de entrada anterior, del evento de entrada actual, donde la información de mapa de bits de cada elemento de información de atributo se usa para indicar si la información de atributo es la misma que la información de atributo correspondiente del evento de entrada anterior; obtener, según la indicación de la información de mapa de bits, información de atributo, que es la misma que la del evento de entrada anterior, del evento de entrada actual a partir de la información de atributo del evento de entrada anterior; y combinar la información de atributo idéntica obtenida y la información de atributo diferente comprendida en los datos transmitidos para formar el evento de entrada actual.

10
15 13. El servidor según la reivindicación 10 u 11, en el que:

15 el procesador (920) está configurado específicamente para: cuando se recibe una notificación que indica que se ha producido un cambio en un estado de sistema de la máquina virtual, determinar si la información de atributo de los múltiples eventos de entrada coincide con el estado de sistema; e insertar un evento de entrada correspondiente a la información de atributo que coincide con el estado de sistema en un sistema de la máquina virtual, de manera que el sistema invoque una función de procesamiento para procesar el evento de entrada.

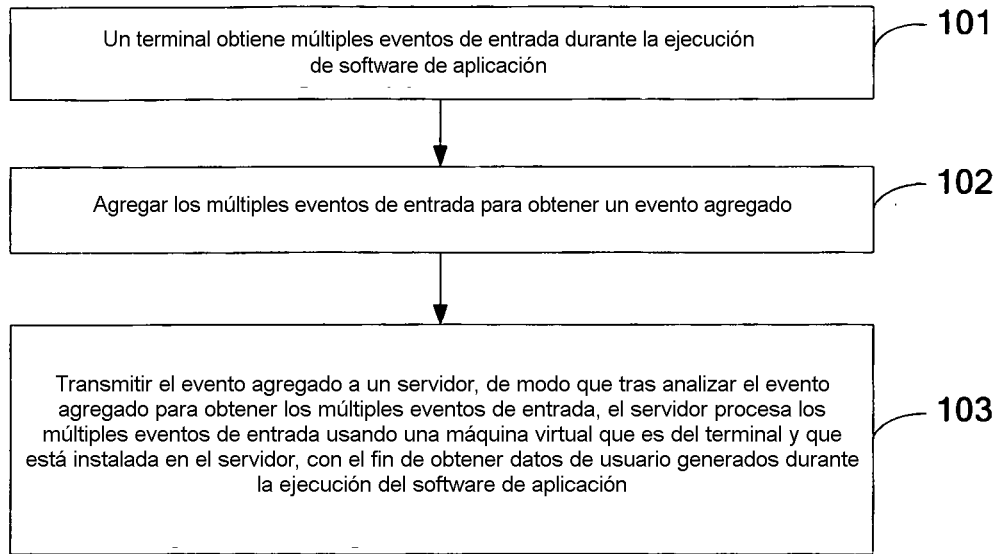


FIG. 1A

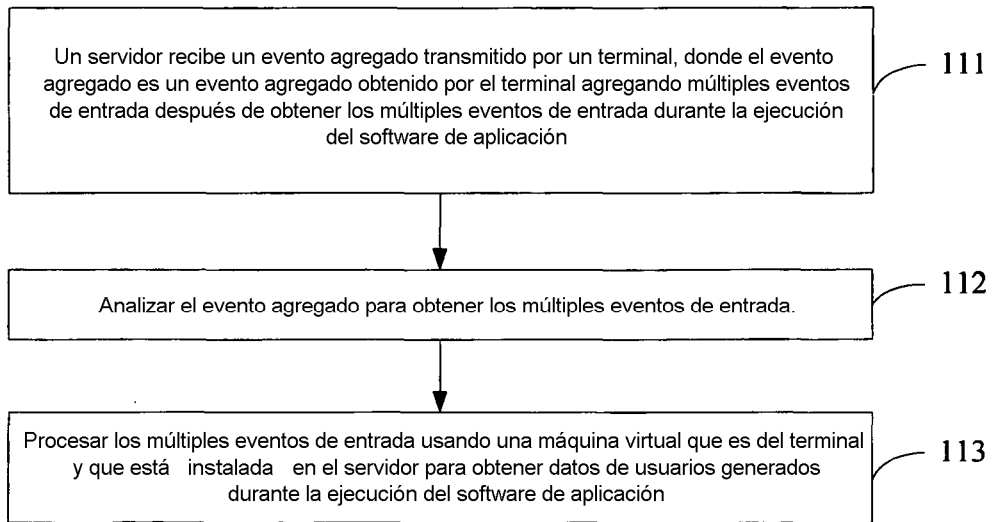


FIG. 1B

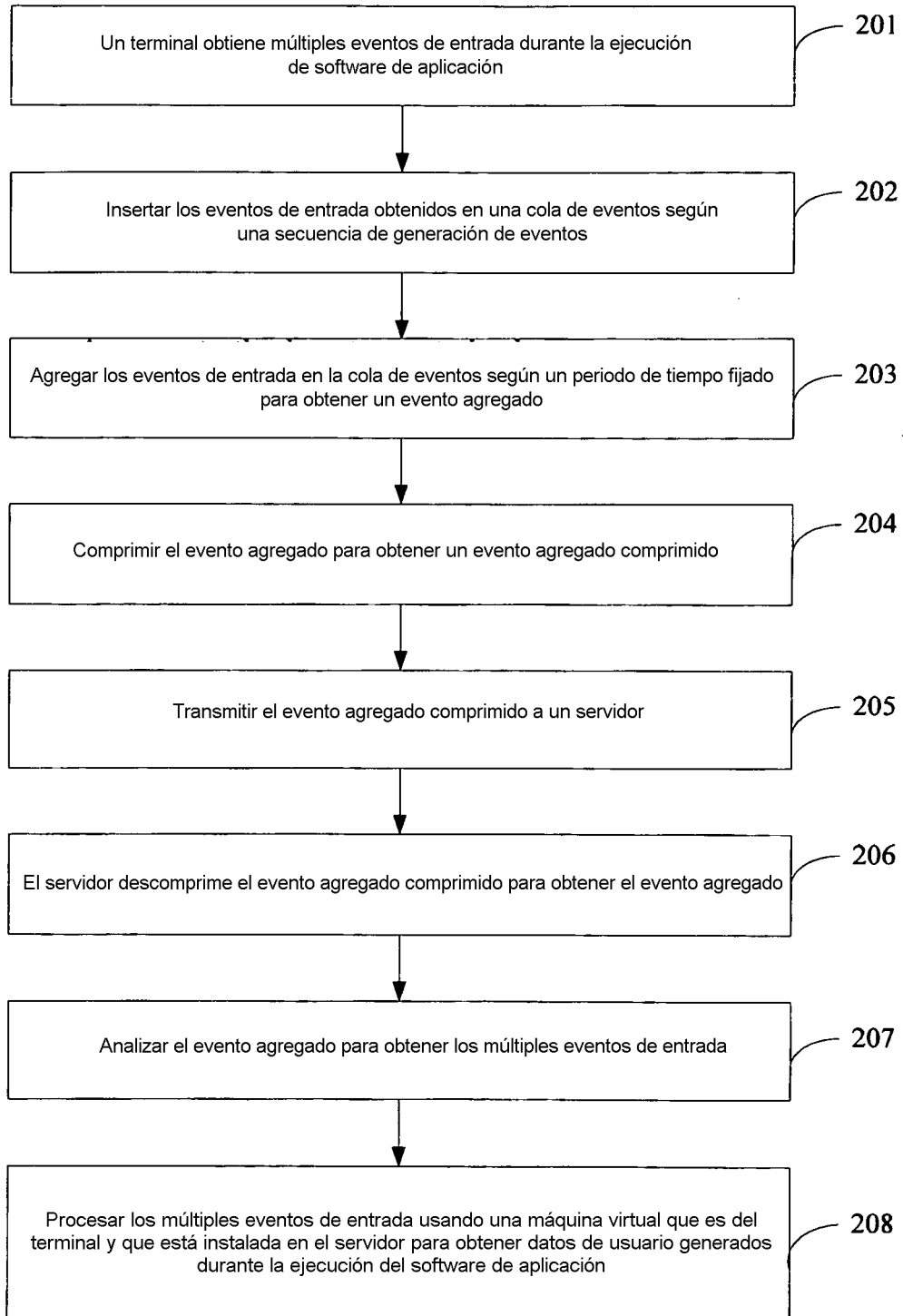


FIG. 2

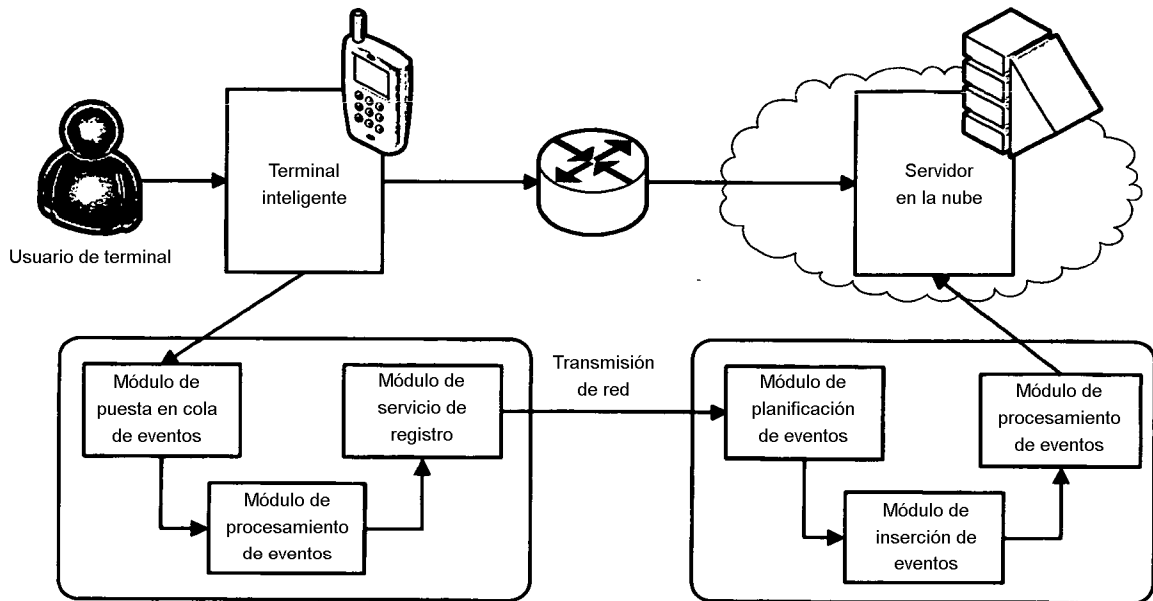


FIG. 3A

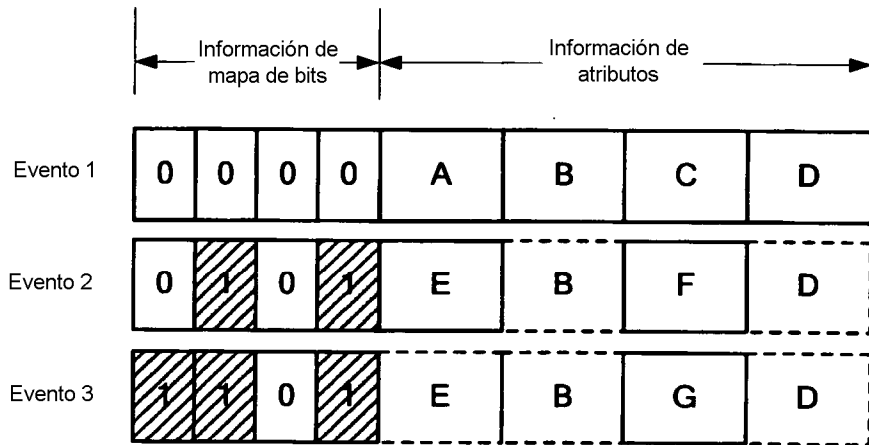


FIG. 3B

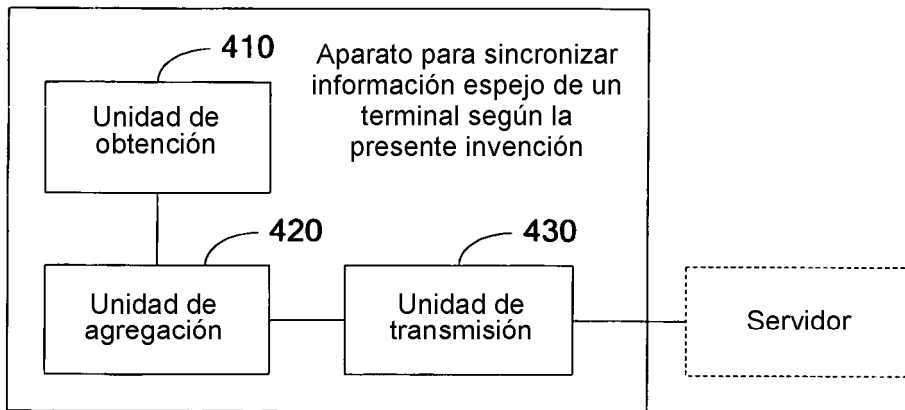


FIG. 4

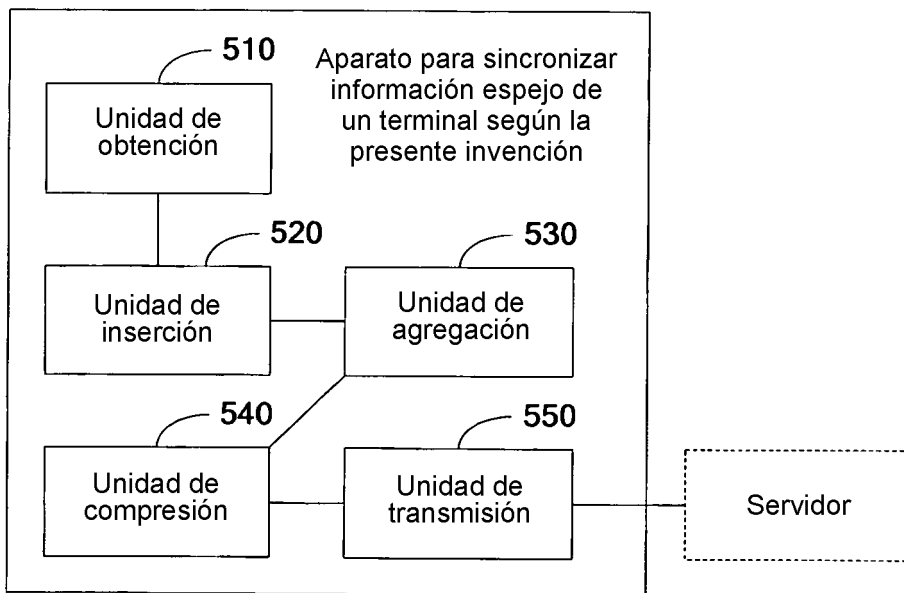


FIG. 5

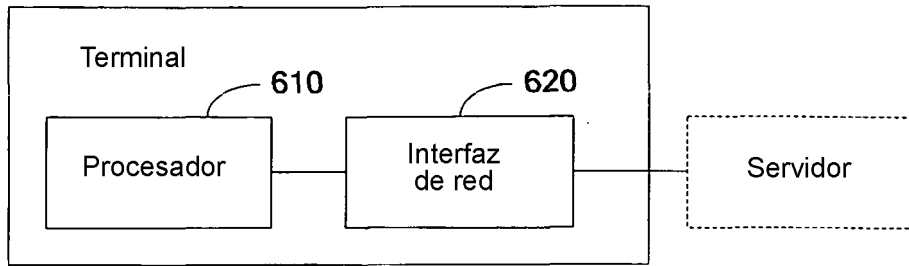


FIG. 6

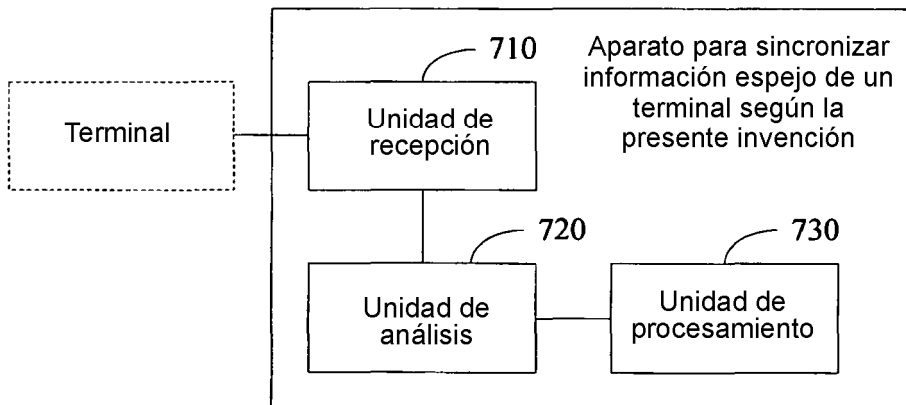


FIG. 7

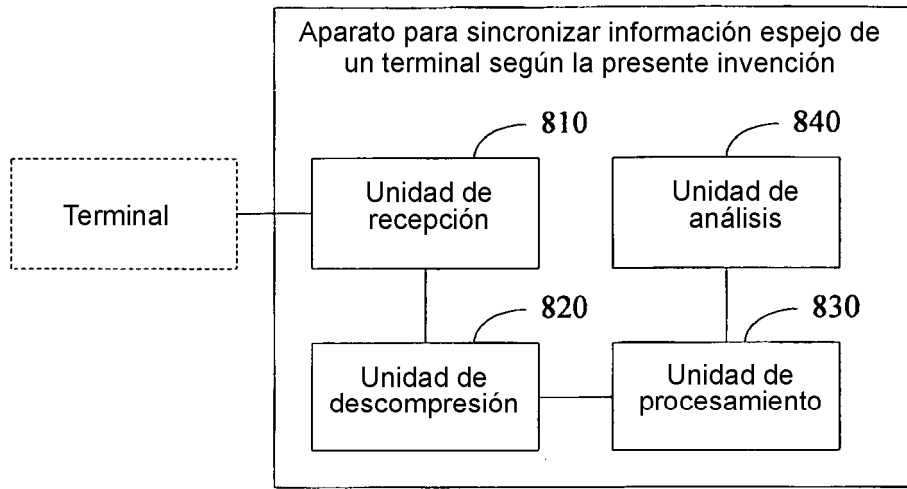


FIG. 8

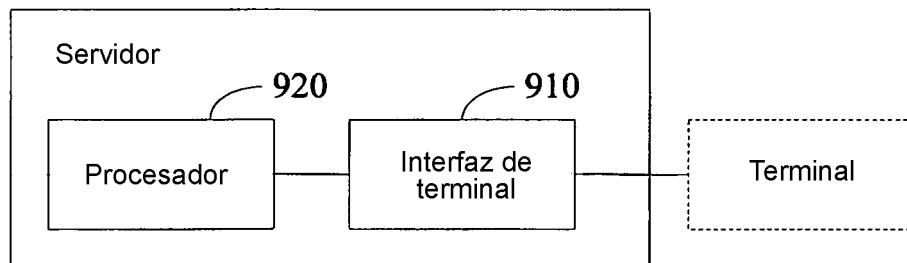


FIG. 9