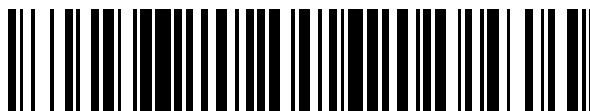


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 652 262**

51 Int. Cl.:

G06F 11/14 (2006.01)

G06F 11/20 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **16.12.2014 PCT/US2014/070456**

87 Fecha y número de publicación internacional: **09.07.2015 WO15102874**

96 Fecha de presentación y número de la solicitud europea: **16.12.2014 E 14828391 (4)**

97 Fecha y número de publicación de la concesión europea: **08.11.2017 EP 3090345**

54 Título: **Método de retardar puntos de comprobación inspeccionando paquetes de red**

30 Prioridad:

30.12.2013 US 201361921720 P

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

01.02.2018

73 Titular/es:

**STRATUS TECHNOLOGIES BERMUDA LTD.
(100.0%)
Canon's Court, 22 Victoria Street
Hamilton HM 12, BM**

72 Inventor/es:

**BISSETT, THOMAS D.;
LEVEILLE, PAUL A. y
CHINTA, SRINIVASU**

74 Agente/Representante:

UNGRÍA LÓPEZ, Javier

ES 2 652 262 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

Método de retardar puntos de comprobación inspeccionando paquetes de red

5 **Campo de la invención**

La invención se refiere en general al campo de la computación tolerante a fallos y, más específicamente, a sistemas de puntos de comprobación, conexión de redes y virtualización.

10 **Antecedentes de la invención**

Hay varias formas de lograr computación tolerante a fallos. Específicamente, se puede usar hardware tolerante a fallos y software tolerante a fallos solos o juntos. Por ejemplo, es posible conectar dos (o más) ordenadores, de tal manera que un ordenador, el ordenador activo o host, haga activamente cálculos mientras que el otro ordenador (u ordenadores) está inactivo o en espera en caso de que el ordenador activo, o su componente de hardware o software, sufra algún tipo de fallo. En estos sistemas, la información acerca del estado del ordenador activo debe ser guardada periódicamente en el ordenador de reserva de modo que el ordenador de reserva pueda asumir sustancialmente las funciones del ordenador previamente activo en el punto de los cálculos donde el ordenador activo experimentó un fallo. Esta función puede ampliarse para uso en la práctica diaria moderna de usar un entorno virtualizado como parte de una nube u otro sistema informático.

La virtualización se usa en muchos campos para reducir el número de servidores u otros recursos necesarios para un proyecto u organización concretos. Los actuales sistemas informáticos de máquina virtual utilizan máquinas virtuales (MV) que operan como huéspedes dentro de un ordenador host físico. Cada máquina virtual incluye su propio sistema operativo virtual y opera bajo el control de un sistema operativo de gestión o hipervisor que se ejecuta en la máquina host física. Cada máquina virtual ejecuta una o varias aplicaciones y accede a redes físicas de almacenamiento de datos y ordenadores según requieran las aplicaciones. Además, cada máquina virtual puede actuar, a su vez, como el sistema informático host para otra máquina virtual.

Múltiples máquinas virtuales pueden estar configuradas como un grupo para ejecutar uno o varios programas idénticos. Típicamente, una máquina virtual del grupo es la máquina virtual primaria o activa, y las máquinas virtuales restantes son las máquinas virtuales secundarias o de reserva. Si algo va mal en la máquina virtual primaria, una de las máquinas virtuales secundarias puede asumir el control y su papel en el sistema de computación tolerante a fallos. Esta redundancia permite que el grupo de máquinas virtuales opere como un sistema de computación tolerante a fallos. La máquina virtual primaria ejecuta aplicaciones, recibe y envía datos de red, y lee y escribe en el almacenamiento de datos mientras realiza tareas o interacciones automatizadas o iniciadas por el usuario. Las máquinas virtuales secundarias tienen las mismas capacidades que la máquina virtual primaria, pero no asumen las tareas y actividades relevantes hasta que la máquina virtual primaria falla o queda afectada por un error.

Para que tal grupo de máquinas virtuales funcione como un sistema tolerante a fallos, el estado operativo, que define el contenido de la memoria y del almacenamiento de datos de una máquina virtual secundaria, deberá ser equivalente al estado operativo que es el contenido de la memoria y el almacenamiento de datos de la máquina virtual primaria. Si se cumple esta condición, la máquina virtual secundaria puede sustituir a la máquina virtual primaria sin pérdida de datos. Para asegurar que el estado de la máquina secundaria y su memoria sea equivalente al estado de la máquina primaria y su memoria, es necesario que la máquina virtual primaria transfiera periódicamente su estado y contenido de memoria a la máquina virtual secundaria.

La transferencia periódica de datos para mantener la sincronía entre los estados de las máquinas virtuales se denomina punto de comprobación. Un punto de comprobación define un punto del tiempo en el que los datos han de ser transferidos. Durante un punto de comprobación, el procesado en la máquina virtual primaria se pone en pausa, de modo que el estado final de la máquina virtual y memoria asociada no se cambia durante el intervalo del punto de comprobación y, una vez que los datos relevantes son transferidos, tanto la máquina primaria como las virtuales secundarias están en el mismo estado. La máquina virtual primaria reanuda entonces su funcionamiento y continúa ejecutando la aplicación hasta el punto de comprobación siguiente, en el que el proceso se repite.

Los puntos de comprobación pueden ser determinados por el transcurso de una cantidad fija de tiempo desde el último punto de comprobación o por la aparición de algún evento, tal como: el número de accesos a memoria (denominados páginas sucias); la aparición de un evento de red (tal como el reconocimiento de red que sale de la máquina virtual primaria); o la aparición de memoria intermedia excesiva en la máquina virtual secundaria (en comparación con la memoria disponible), durante la ejecución de la aplicación. El punto de comprobación de tiempo transcurrido se considera un punto de comprobación fijo, mientras que un punto de comprobación basado en evento se considera un punto de comprobación dinámico o de tasa variable.

Los puntos de comprobación excesivos pueden dar lugar a degradación del rendimiento de la máquina virtual primaria. A su vez, esta degradación del rendimiento puede dar lugar a retardos y pérdida de datos, lo que puede

poner en peligro la naturaleza de tolerancia a los fallos del sistema. La activación de puntos de comprobación en respuesta al tráfico de red puede ser especialmente gravosa para un sistema de puntos de comprobación.

Por lo tanto, se necesitan formas de reducir la carga general del sistema sin reducir la fiabilidad del sistema.

Las realizaciones de la invención afrontan esta necesidad y otras.

US 2007/0174484 describe un sistema informático en el que se fuerza la realización de un intervalo de punto de comprobación cuando expira un temporizador de paquetes diferidos. US 2010/107158 describe la tolerancia a fallos de baja utilización de recursos mediante puntos de comprobación híbridos y repetición. US 2011/167196 describe proporcionar tolerancia a los fallos de una máquina virtual primaria transmitiendo de forma continua información de punto de comprobación a un proceso de recogida.

Resumen de la invención

La invención proporciona un método de retardar los puntos de comprobación de tasa variable, y un sistema de computación tolerante a fallos, como los expuestos en las reivindicaciones independientes anexas 1 y 12.

Las realizaciones de la invención se refieren a un método para retardar los puntos de comprobación en un sistema de máquina virtual. En una realización, el método incluye los pasos de determinar si una trama es diferible; y si la trama es una trama diferible, retardar un punto de comprobación asociado con la trama. En otra realización la trama es examinada para determinar si es una trama diferible. En otra realización, la trama diferible es un paquete etiquetado con uno o más de: no es el último fragmento de datos de un mensaje de fragmentos múltiples; un segmento de datos que no indica una descarga de la memoria intermedia de envío; y un segmento que solamente sirve como un reconocimiento de algún mensaje anterior. En otra realización, la trama diferible es un paquete IP etiquetado con uno o más de: un atributo de 'más fragmentos'; un segmento de datos TCP que carece del señalizador PSH y que no lleva señalizadores distintos de 'ACK'; y un segmento TCP que no contiene datos y lleva solamente el señalizador 'ACK'. En otra realización, la trama diferible es identificada por el protocolo en uso.

En una realización, la trama diferible es identificada por el estado de sesión del protocolo en uso. En otra realización, la trama diferible es identificada por el número de puerto. En otra realización, el número de puerto es un número de puerto fuente. En otra realización, el número de puerto es un número de puerto destino. En otra realización, la trama diferible es identificada por una dirección de red. En una realización, la dirección de red es una dirección de red fuente. En otra realización, la dirección de red es una dirección de red destino.

En una realización, el método incluye el paso de concatenar los retardos de punto de comprobación debido a una pluralidad de tramas diferibles. En otra realización, el método incluye el paso de establecer un límite superior a la cantidad de retardo que puede generarse. En otra realización, se determina que la trama es diferible en base a la interfaz de red en uso.

Las realizaciones de la invención también se refieren a un sistema de computación tolerante a fallos. En una realización, el sistema informático tolerante a fallos incluye un primer dispositivo informático incluyendo una máquina virtual primaria; y un segundo dispositivo informático incluyendo una máquina virtual secundaria, donde el primer dispositivo informático y el segundo dispositivo informático están en red, donde la máquina virtual primaria incluye un primer motor de punto de comprobación y una primera interfaz de red, donde la máquina virtual secundaria incluye un segundo motor de punto de comprobación y una segunda interfaz de red, y donde el primer motor de punto de comprobación retarda un punto de comprobación para una o más tramas diferibles de datos. En otra realización, el primer motor de punto de comprobación pausa la máquina virtual primaria cuando se declara un punto de comprobación. En otra realización, el primer motor de punto de comprobación añade un retardo adicional antes de declarar un punto de comprobación posterior. En otra realización, el primer motor de punto de comprobación captura el estado de la máquina virtual primaria mientras la máquina virtual primaria está en pausa.

Breve descripción de los dibujos

La estructura y la función de la invención se pueden entender mejor a partir de esta descripción en unión con las figuras acompañantes. Las figuras no están necesariamente a escala, recalándose en cambio en general los principios ilustrativos. Las figuras se han de considerar ilustrativas en todos los aspectos y no tienen la finalidad de limitar la invención, cuyo alcance se define solamente por las reivindicaciones.

La figura 1 es un diagrama de bloques de una realización de un motor de punto de comprobación en una máquina virtual construida según la invención.

Las figuras 2 (a) y (b) son diagramas de tiempo conocidos a la técnica anterior para una realización de una máquina virtual y un motor de punto de comprobación, respectivamente.

Las figuras 3 (a) y (b) son diagramas de tiempo para una realización de una máquina virtual y un motor de punto de comprobación, respectivamente, construidos según la invención.

Descripción de una realización preferida

Aquí se describen realizaciones detalladas de la invención; sin embargo, se ha de entender que las realizaciones descritas son simplemente ejemplares de la invención, que puede realizarse de varias formas. Por lo tanto, los detalles funcionales específicos aquí descritos no se han de interpretar como limitativos, sino simplemente como una base para las reivindicaciones y como una base representativa para mostrar a los expertos en la técnica cómo emplear de varias formas la invención virtualmente en cualquier realización apropiadamente detallada.

En un sistema de punto de comprobación de tasa variable, puede producirse una tasa excesiva de puntos de comprobación cuando se inician ciclos de punto de comprobación en la máquina virtual primaria en respuesta a la aparición de un evento en la máquina virtual primaria tal como transmisión o tráfico de datos de red de salida. Normalmente, tal tráfico de red de salida produce un ciclo inmediato de puntos de comprobación para asegurar la aparición de intercambios de latencia más baja entre la máquina virtual primaria y el ordenador en la red que recibe la transmisión de la máquina virtual. Éste es de hecho el caso para operaciones a nivel de archivo tales como enumeración de carpetas, borrado de archivos, manipulación de atributos, e incluso intercambios de transacciones de hilo único.

Bajo estos tipos de intercambios sensibles a latencia, es deseable tener una tasa de puntos de comprobación rápida y sensible. Los intercambios sensibles a latencia, por ejemplo, algunas peticiones de cliente y respuestas de servidor, se benefician de un mecanismo de punto de comprobación muy sensible.

Sin embargo, otros intercambios, tales como algunos señalizadores y parámetros de protocolo de comunicación, entre un cliente y servidor son de tipo de "mantenimiento interno". Las comunicaciones de esta actividad del tipo de mantenimiento interno o de procedimiento producen tráfico de salida de la máquina virtual primaria que da lugar a excesiva actividad de punto de comprobación. Esta excesiva actividad de punto de comprobación es perjudicial para la eficiencia de la aplicación y requiere el manejo de recursos sustanciales de procesado y comunicación entre hosts.

Un ejemplo de un intercambio de datos de red que es del tipo de mantenimiento interno es un simple paquete de reconocimiento TCP o 'ACK'. Cuando un cliente está transfiriendo un archivo a la máquina virtual huésped usando protocolo TCP, la pila TCP de la máquina virtual huésped responde típicamente a cada mensaje/segmento con un 'ACK'. Pueden generarse cientos de tales 'ACKs' rápidos cuando dicha corriente entrante de segmentos está en curso. El ordenador que recibe el ACK no depende del ACK con el fin de seguir operando o manejando otras tareas de procesado necesarias. Además, con pocas excepciones, no es necesario que el ordenador cliente reciba estos 'ACKs' en tiempo real. La realización de un ciclo inmediato de puntos de comprobación por cada 'ACK' interferiría en gran medida con el flujo entrante de mensajes/segmentos de tramas de cliente proporcionando al mismo tiempo poco o nulo beneficio.

Otro ejemplo de tal intercambio de datos de red de mantenimiento interno es un paquete IP etiquetado con el señalizador IP_MF, que indica que más fragmentos de paquete de datos están entrando. Estos tipos de tramas de salida aumentan la tasa de puntos de comprobación y carga del sistema asociado, dando lugar por ello a una eficiencia general más baja del sistema. Como resultado, los puntos de comprobación que surgen de estos tipos de intercambios de datos de red no son críticos y puede diferirse para procesado en un tiempo posterior.

Según una realización, el tráfico de red de salida y los parámetros operativos pueden ser supervisados para detectar datos tales como tramas que pueden ser clasificadas como diferibles. Por diferible se entiende que el ordenador que recibe los datos, como un ACK, no espera típicamente la trama específica con el fin de seguir procesando u operando de otro modo. Los ejemplos de datos diferibles incluyen: paquetes IP etiquetados con el atributo 'más fragmentos'; segmentos de datos TCP que carecen del señalizador PSH (forzar envío) y no llevan señalizadores distintos de 'ACK'; y segmentos TCP que no contienen datos y solamente contienen el señalizador 'ACK'. Puede haber otras tramas y tipos de datos candidato y son manejados de forma similar.

Las realizaciones anteriores describen paquetes específicos TCP/IP y definiciones de campo que pueden generalizarse a casi cualquier protocolo de red. Los paquetes de mantenimiento interno solamente, tales como "ACK" específicos de IP, son candidatos a paquetes diferibles. Los paquetes fragmentados, tales como los identificados en IP con IP_MF, son candidatos a paquetes diferibles. Igualmente, los paquetes que no contienen datos y que no son parte de una descarga de memoria intermedia de envío a nivel (como en el señalizador TCP PSH), también son candidatos a paquetes diferibles. Otros muchos tipos de paquete y condiciones pueden ser identificados dependiendo del protocolo específico que se use. Hay cientos de protocolos de red registrados, cada uno de los cuales tiene sus propias características de comportamiento. Cada paquete en un protocolo dado puede ser diferible.

Con referencia a la figura 1 se muestran componentes de un motor de punto de comprobación 122 con una máquina virtual 100. La máquina virtual 100 estaría conectada típicamente a una segunda máquina virtual (no representada) para lograr un sistema de computación tolerante a fallos. La máquina virtual 100 incluye un módulo de punto de comprobación 101. El módulo de punto de comprobación 101 pone en memoria intermedia datos de red 109 en un módulo de entrada/salida (E/S) 110 y un evento de notificación 112 alerta al motor de punto de comprobación 122. Los datos almacenados (referidos genéricamente como bloque E/S) se pueden mover 138 más libremente entre el módulo de entrada/salida (E/S) 110 y la interfaz de bloque 114. El motor de punto de comprobación 122 está en comunicación con un módulo de seguimiento de memoria 126 y un cache intermedia 130. Cuando el motor de punto de comprobación 122 está inactivo, los datos de red (si los hay) permanecen en un estado de puesta en memoria intermedia por el módulo de entrada/salida (E/S) 110, y solamente serán liberados por la señal 113 al módulo de entrada/salida (E/S) 110 después de que el punto de comprobación siguiente llegue a su etapa de realización/liberación. En ese momento, los datos de red (si los hay) son transferidos 134 del módulo de entrada/salida (E/S) 110 a la interfaz de red 118. Cuando se declara un punto de comprobación, la máquina virtual 100 se pausa y los datos de páginas sucias son procesados por el motor de punto de comprobación 122. En el caso de la presente invención, cuando se declara un punto de comprobación, por ejemplo, por una trama ACK recibida por el módulo de E/S 110, la trama está disponible para el motor de punto de comprobación 122 para inspección.

Con más detalle y con referencia a la figura 2(a), un diagrama de tiempo de los eventos y métodos que tienen lugar en secuencia para una máquina virtual primaria se representa para un sistema de puntos de comprobación conocido en la técnica anterior. La máquina virtual primaria está funcionando (ejecutando la aplicación) 210, 210' o en pausa 214. El motor de punto de comprobación incluye condiciones de tiempo adicionales como se representa en la figura 2(b). El motor de punto de comprobación puede estar inactivo 220 hasta que tiene lugar 224 un punto de comprobación (P), tiempo en el que el motor de punto de comprobación hace 226 que la máquina virtual primaria se pause 214. Mientras la máquina virtual primaria está en pausa 214, el estado de la máquina virtual primaria es capturado 228 por el motor de punto de comprobación. La captura es una etapa de copia de página que permite la reanudación y ejecución de la máquina virtual en paralelo con 'transferencia' de punto de comprobación y 'esperar ACK'.

En otra realización, se evita la copia de páginas y las páginas son transferidas tal cual. En este caso, el huésped permanece pausado durante las etapas de transferencia y espera de ACK. También pueden usarse combinaciones de estos dos acercamientos. Una vez que los datos de estado son capturados 228, el motor de punto de comprobación reanuda 232 la máquina virtual primaria que de nuevo ejecuta 210' la aplicación.

Los datos de estado capturados son transferidos posteriormente 236 a la máquina virtual secundaria. Una vez que los datos han sido transferidos, el motor de punto de comprobación espera que se reciba 240 un reconocimiento (ACK) de la máquina virtual secundaria, indicando que los datos han sido recibidos. En este punto, se ocasiona el ciclo de puntos de comprobación, y los datos de red puestos en memoria intermedia son liberados 244 a la red. En este punto, el motor de punto de comprobación entra en estado inactivo 220' hasta que tiene lugar el punto de comprobación siguiente 224'.

Cuando tienen lugar algunos eventos de activación, que no son diferibles, se declara un punto de comprobación. Algunos ejemplos no limitadores de eventos de activación de punto de comprobación son los siguientes: un temporizador que inicia un período de punto de comprobación, un evento de red. Tal como que se requiere una respuesta ACK desde la máquina virtual primaria a otro ordenador en la red, y que tiene lugar una operación de escritura en una página de memoria. Aunque se declare un punto de comprobación, hasta que la máquina virtual sea puesta en pausa por el motor de punto de comprobación, puede haber eventos de generación de puntos de comprobación adicionales. Así, muchos puntos de comprobación pueden estar en cola con cada acción requerida por el motor de punto de comprobación. Por ejemplo, cada punto de comprobación posterior requerirá, como mínimo, una pausa y vuelta a arrancar de la máquina virtual, contribuyendo a la carga de tiempo del sistema en el que la máquina virtual primaria no está realizando ningún trabajo útil.

Cuando se detecta una trama diferible u otros datos diferibles, tales como un ACK, el sistema hace que un punto de comprobación sea programado en el tiempo en algún punto futuro próximo, en vez de generarse inmediatamente. En algunas realizaciones, las tramas diferibles posteriores pueden modificar el temporizador, cuando sea apropiado, para ampliar, acortar o cancelar el retardo al ciclo programado de puntos de comprobación. El temporizador de retardo es cancelado automáticamente cuando se detectan datos no diferibles, o tiene lugar otro evento de activación para un punto de comprobación (que invalida el retardo de un punto de comprobación), volviendo por ello a la política normal de iniciación de punto de comprobación en ese instante.

La cantidad de retardo de trama diferible puede ser fijo o calcularse en base a otras entradas, pero deberá ser suficientemente grande para permitir que el sistema agregue otras tramas de datos diferibles que probablemente aparecerán dentro del retardo. Sin embargo, el retardo también deberá ser suficientemente corto para evitar problemas en el caso de que una trama considerada como diferible sea, por alguna razón, sensible en el tiempo al ordenador receptor. En una realización, es adecuado un retardo de aproximadamente 5 a aproximadamente 10 milisegundos.

Con referencia a la figura 3(a), la temporización de la máquina virtual se representa con respecto a un sistema de puntos de comprobación según la presente invención. En esta realización, la máquina virtual primaria está funcionando (ejecutando una o varias aplicaciones) 310, 310' o en pausa 314. El motor de punto de comprobación opera en las condiciones de tiempo, como se representa en la figura 3(b). Un número de transiciones de tiempo de motor de punto de comprobación es el mismo que el representado en la figura 2(b). Por ejemplo, el motor de punto de comprobación puede estar inactivo 320 hasta que tiene lugar 324 un punto de comprobación (P), tiempo en el que el motor de punto de comprobación hace 326 que la máquina virtual primaria se pause 314. Mientras la máquina virtual primaria está en pausa 314, el estado de la máquina virtual primaria es capturado 328 por el motor de punto de comprobación. Una vez que los datos de estado son capturados 328, el motor de punto de comprobación reanuda 332 la máquina virtual primaria que de nuevo ejecuta 310' una o varias aplicaciones.

Los datos de estado capturados son transferidos posteriormente 336 a la máquina virtual secundaria. Una vez que los datos han sido transferidos, el motor de punto de comprobación espera que se reciba un reconocimiento (ACK) 340 de la máquina virtual secundaria, indicando que los datos han sido recibidos. En este punto, los datos de punto de comprobación son realizados 2344, por ejemplo, liberando una o varias tramas de datos a la red.

El motor de punto de comprobación entra entonces en un estado inactivo 348, pero, en esta realización, el estado inactivo incluye un período de retardo adicional si los eventos que generan un punto de comprobación posterior son diferibles. Si los eventos son diferibles, se añade un retardo al período de inactividad antes de que se declare el punto de comprobación siguiente 324'. Este retardo incrementado permite que eventos de puntos de comprobación adicionales estén en cola antes de que se declare un punto de comprobación, permitiendo así cubrir múltiples eventos de punto de comprobación por un conjunto de operaciones de punto de comprobación. Esto reduce la carga por evento para manejar un punto de comprobación.

El método para determinar la existencia de una trama diferible deberá aplicarse de forma fácil y rápida, sin necesidad de rastrear la historia de una conexión/sesión. En este sentido, la prueba para una trama diferible no tiene estado con respecto al diálogo en curso entre la máquina virtual y el otro ordenador en la red. Algunas características ejemplares de implementar un método basado en datos diferibles y retardo de punto de comprobación son: actuar en los protocolos de transporte de uso más común (pero el método también puede adaptarse a protocolos personalizados); no hacer que las tramas caigan o sean reordenadas como un medio de mejorar el rendimiento de punto de comprobación; y controlar la selectividad aplicando diferentes reglas a varios tipos de tramas diferibles.

Además, el retardo de punto de comprobación puede aplicarse no solamente según que la trama propiamente dicha sea diferible, sino también si la trama está dirigida o se origina a partir de direcciones de red específicas (MAC, IP, etc) y/o números de puertos, de los que algunos se usan típicamente por convención para algunas aplicaciones. Así, el sistema puede prepararse para que difiera cualquier paquete (o evite el diferimiento y el retardo de punto de comprobación de cualquier paquete) que se origine en o vaya destinado a una dirección de red específica y/o un número de puerto (por ejemplo, el puerto 89) en base a convención o reglas dadas al motor de punto de comprobación. Puede usarse diverso manejo especial de datos importantes para un ordenador en red con el fin de anular diferibles puntos de comprobación y retardos de punto de comprobación o para producir de forma más agresiva retardos relativos a direcciones de red y/o puertos seleccionados.

En otra realización, la máquina virtual tiene múltiples interfaces de red. El tráfico de red en una interfaz de red específica puede ser de una naturaleza tal que todos sus paquetes sean diferibles. También es posible que una interfaz de red específica solamente lleve paquetes no diferibles.

Un ejemplo de manejo de casos especiales en el contexto de puntos de comprobación es TCP ACKs conocido como 'actualizaciones de ventana'. Estos ACKs proporcionan información específica acerca de la ventana de envío TCP del emisor, y su retardo puede hacer ocasionalmente que el emisor (el cliente) pause su transmisión. Detectar exactamente cuándo la información de 'actualización de ventana' es importante para el emisor es difícil y mejora poco el acercamiento general de gestión de tramas diferibles, especialmente cuando el tamaño de la ventana de envío llega a un tamaño pleno de 64KB (típico) o mayor (escala de ventana).

Sin embargo, todavía pueden producirse tales paradas de envío de ventana, lo que es una razón importante para impedir que el retardo diferible sea demasiado grande. Así, es importante limitar la cantidad de retardo que puede generarse, así como no permitir que el retardo afecte a otras funciones de la máquina virtual. También es aceptable usar seguimiento parcial o pleno del estado de sesión para no retardar selectivamente un paquete de "actualización de ventana".

A no ser que se indique específicamente lo contrario cuando sea evidente por la explicación siguiente, se hace notar que, en toda la descripción, las explicaciones que utilizan términos como "procesado" o "computación" o "calcular" o "retardar" o "comparar", "generar" o "determinar" o "diferir" o "ocasionar" o "punto de comprobación" o "interrupción" o "manejo" o "recepción" o "puesta en memoria intermedia" o "asignar" o "visualizar" o "señalizar" o lógica booleana u otras operaciones relacionadas o análogos, se refieren a la acción y los procesos de un sistema informático, o dispositivo electrónico, que manipula y transforma datos representados como cantidades físicas (electrónicas) dentro

de los registros y memorias del sistema informático o los dispositivos electrónicos a otros datos igualmente representados como cantidades físicas dentro de memorias o registros electrónicos u otros dispositivos de almacenamiento, transmisión o visualización de tal información.

5 Los algoritmos aquí presentados no están inherentemente relacionados con cualquier ordenador u otro aparato concreto. Pueden usarse varios sistemas generales con programas según las ideas de la invención, o puede demostrarse que es conveniente construir un aparato más especializado para realizar los pasos requeridos del método. La estructura requerida para una variedad de estos sistemas se entenderá por la descripción anterior. Además, la presente invención no se describe con referencia a ningún lenguaje de programación concreto, y varias realizaciones pueden implementarse así usando varios lenguajes de programación.

10 Los aspectos, las realizaciones, las características y los ejemplos de la invención se han de considerar ilustrativos en todos los aspectos y no tienen la finalidad de limitar la invención, cuyo alcance se define solamente por las reivindicaciones. Otras realizaciones, modificaciones y usos serán evidentes a los expertos en la técnica sin apartarse del alcance de la invención reivindicada.

15 En la solicitud, donde se afirma que un elemento o componente está incluido en y/o seleccionado de una lista de elementos o componentes expuestos, se deberá entender que el elemento o componente puede ser alguno de los elementos o componentes expuestos y se puede seleccionar de un grupo que consta de dos o más de los elementos o componentes expuestos. Además, se deberá entender que elementos y/o características de una composición, un aparato, o un método aquí descritos pueden combinarse de varias formas sin apartarse del alcance de las ideas de la presente invención, explícitas o implícitas.

20 El uso de los términos “incluir”, “incluye”, “incluyendo”, “tener”, “tiene” o “teniendo” deberá entenderse en general en sentido abierto y no limitativo a no ser que se indique específicamente lo contrario.

25 Se deberá entender que el orden de los pasos o el orden para realizar ciertas acciones es intrascendente a condición de que las ideas de la presente invención sigan siendo operables. Además, dos o más pasos o acciones pueden realizarse simultáneamente.

30 Se ha de entender que las figuras y las descripciones de la invención se han simplificado para ilustrar elementos que son relevantes para una clara comprensión de la invención, mientras que se eliminan, para claridad, otros elementos. Los expertos en la técnica reconocerán, sin embargo, que estos y otros elementos pueden ser deseables. Sin embargo, dado que tales elementos son conocidos en la técnica, y dado que no facilitan una mejor comprensión de la invención, no se ofrece aquí una explicación de tales elementos. Se deberá apreciar que las figuras se presentan a efectos ilustrativos y no como dibujos de construcción. Los detalles omitidos y las modificaciones o realizaciones alternativas caen dentro del alcance de las personas de conocimientos ordinarios en la técnica.

35 La invención puede realizarse en otras formas específicas sin apartarse de sus características esenciales. Por lo tanto. Las realizaciones anteriores se han de considerar en todos los aspectos como ilustrativas más bien que como limitaciones de la invención aquí descrita. El alcance de la invención se indica así en las reivindicaciones anexas más bien que en la descripción anterior.

45

REIVINDICACIONES

- 5 1. Un método de retardar los puntos de comprobación de tasa variable en una máquina virtual (100) de un sistema de computación tolerante a fallos dispuesto para iniciar ciclos de punto de comprobación en respuesta a transmisión de tramas de datos de red de salida, incluyendo los pasos de:
- 10 determinar si una trama para transmisión de red de salida es una trama diferible, donde una trama diferible quiere decir que un sistema informático que recibe la trama no espera típicamente a la trama específica para continuar el procesado u operar de otro modo; y
- 10 si la trama es una trama diferible, retardar un punto de comprobación asociado con la trama.
- 15 2. El método de la reivindicación 1, donde la trama es examinada para determinar si es una trama diferible.
- 15 3. El método de la reivindicación 2, donde la trama diferible es un paquete etiquetado con uno o varios de:
- 20 no ser el último fragmento de datos de un mensaje de fragmentos múltiples;
- 20 un segmento de datos que no indica una descarga de la memoria intermedia de envío; y
- 20 un segmento que solamente sirve como un reconocimiento de algún mensaje anterior.
- 25 4. El método de la reivindicación 2, donde la trama diferible es un paquete IP etiquetado con uno o varios de:
- 25 un atributo de 'más fragmentos';
- 30 un segmento de datos TCP que carece del señalizador PSH y que no lleva señalizadores distintos de 'ACK'; y
- 30 un segmento TCP que no contiene datos y lleva solamente el señalizador 'ACK'.
- 35 5. El método de la reivindicación 2, donde la trama diferible es identificada por el protocolo en uso, opcionalmente por el estado de sesión del protocolo en uso.
- 35 6. El método de la reivindicación 2, donde la trama diferible es identificada por el número de puerto.
- 40 7. El método de la reivindicación 6, donde el número de puerto es un número de puerto fuente o un número de puerto destino.
- 40 8. El método de la reivindicación 2, donde la trama diferible es identificada por la dirección de red.
- 45 9. El método de la reivindicación 8, donde la dirección de red es una dirección de red fuente o una dirección de red destino.
- 45 10. El método de la reivindicación 1, donde se determina que la trama es diferible en base a la interfaz de red en uso.
- 50 11. El método de cualquier reivindicación precedente incluyendo además el paso de concatenar los retardos de punto de comprobación debidos a una pluralidad de tramas diferibles, e incluyendo además opcionalmente el paso de poner un límite superior a la cantidad de retardo que puede ser generada.
- 50 12. Un sistema de computación tolerante a fallos incluyendo:
- 55 un primer dispositivo informático incluyendo una máquina virtual primaria (100); y
- 55 un segundo dispositivo informático incluyendo una máquina virtual secundaria,
- 60 donde el primer dispositivo informático y el segundo dispositivo informático están en red,
- 60 donde la máquina virtual primaria incluye una primera interfaz de red (108), y un primer motor de punto de comprobación de tasa variable (101) dispuesto para iniciar ciclos de punto de comprobación en respuesta a la transmisión de tramas de datos de red de salida mediante la primera interfaz de red;
- 65 donde la máquina virtual secundaria incluye una segunda interfaz de red y un segundo motor de punto de comprobación de tasa variable;

donde el primer motor de punto de comprobación está dispuesto para retardar un punto de comprobación para una o varias tramas de datos para transmisión de red de salida si se determina que dichas tramas de datos son diferibles, donde una trama diferible quiere decir que un ordenador que recibe la trama no espera típicamente a la trama específica para continuar el procesado u operar de otro modo.

5 13. El sistema de computación tolerante a fallos de la reivindicación 12, donde el primer motor de puntos de comprobación está dispuesto para pausar la máquina virtual primaria cuando se declara un punto de comprobación.

10 14. El sistema de computación tolerante a fallos de la reivindicación 13, donde el primer motor de puntos de comprobación está dispuesto para añadir un retardo adicional antes de declarar un punto de comprobación posterior.

15 15. El sistema de computación tolerante a fallos de la reivindicación 12, donde el primer motor de punto de comprobación está dispuesto para capturar el estado de la máquina virtual primaria mientras la máquina virtual primaria está en pausa.

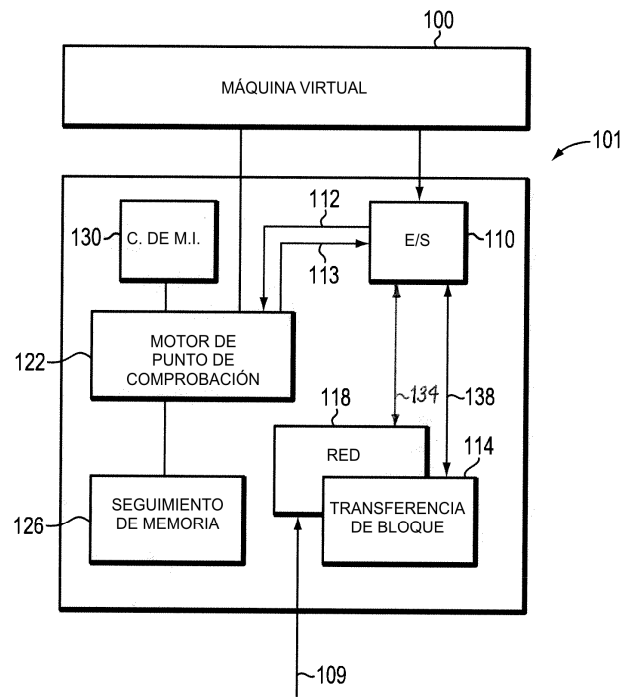


FIG. 1

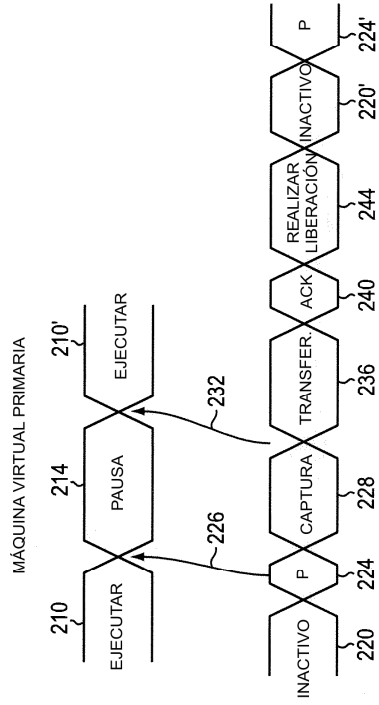


FIG. 2(a)

FIG. 2(b)

