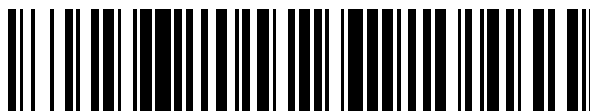


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 555 578**

51 Int. Cl.:

G06F 9/54 (2006.01)

G06F 9/48 (2006.01)

G06F 9/50 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **14.03.2012 E 12708349 (1)**

97 Fecha y número de publicación de la concesión europea: **16.09.2015 EP 2774036**

54 Título: **Aparato, método y programa informático para realizar una migración de un proceso de detector de eventos operativos**

30 Prioridad:

31.10.2011 WO PCT/EP2011/069159

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

05.01.2016

73 Titular/es:

**FRAUNHOFER GESELLSCHAFT ZUR
FÖRDERUNG DER ANGEWANDTEN
FORSCHUNG E.V. (100.0%)
Hansastraße 27 c
80686 München, DE**

72 Inventor/es:

**MUTSCHLER, CHRISTOPHER y
PHILIPPSEN, MICHAEL**

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 555 578 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

Aparato, método y programa informático para realizar una migración de un proceso de detector de eventos operativos

5 Formas de realización de la presente invención se refieren, en general, a redes de datos y, en particular, a aparatos y métodos para la transferencia o migración de detectores de eventos operativos entre diferentes nodos de un sistema informático distribuido.

10 ANTECEDENTES DE LA INVENCION

Redes de detectores, tales como, a modo de ejemplo, redes de detectores inalámbricos tienen una amplia gama de aplicaciones. A modo de ejemplo, redes de sensores inalámbricos de varias tecnologías pueden utilizarse para fines de localización, tal como la localización de personas y/o otros objetos. En este caso, el término "localización" 15 significa la detección o determinación de una posición o situación geográfica. Algunos sistemas de seguimiento de posiciones o localizaciones especializados pueden utilizarse para localizar jugadores y otros objetos (p.ej., un balón) en acontecimientos deportivos, tales como, a modo de ejemplo, fútbol, fútbol americano, rugby, tenis, etc.

20 Cuando se utilizan datos de posicionamiento o localización geográfica agrupados de jugadores y/o un balón, es posible derivar información estadística relacionada con el acontecimiento deportivo completo, a modo de ejemplo, un partido de fútbol o relacionada con equipos o jugadores individuales. Dicha información estadística derivada puede ser de interés por varios motivos. Por un lado, existen varios intereses comerciales como ciertas estadísticas y su análisis puede ser de importancia particular para espectadores en un estadio y/o frente a un aparato de televisión en su domicilio. Por consiguiente, proporcionar algunas estadísticas puede ser de más interés en los acontecimientos 25 deportivos. Por otro lado, los datos estadísticos derivados de los datos de posicionamiento sin procesar pueden utilizarse también para fines de formación profesional. En este caso, un oponente y/o el comportamiento de propio equipo pueden analizarse así como el rendimiento y/o estado de salud de jugadores individuales.

Los sistemas de seguimiento de posiciones o localización antes citados pueden basarse en varias tecnologías. A modo de ejemplo, una información de localización puede determinarse sobre la base de la evaluación de señales de radio inalámbricas y/o campos magnéticos. Para esta finalidad, los transmisores y/o receptores, generalmente 30 indicados también como sensores, pueden colocarse en los objetos individuales (p.ej., jugadores, balón, etc.) para su localización por el sistema. Los dispositivos de recepción y/o transmisión correspondientes pueden montarse también en localizaciones predeterminadas que rodean a una zona geográfica de interés, tal como, a modo de ejemplo, un campo de fútbol. Una evaluación de las intensidades de señales, tiempo de propagación de señales y/o fases de señales, por nombrar tan solo unas pocas alternativas técnicas posibles, pueden llevarse luego a flujos de datos de sensores indicativos de la posición geográfica de objetos o jugadores individuales en diferentes instantes en el tiempo. En condiciones normales, una muestra de datos de localización geográfica está asociada con una marca temporal que indica en qué momento un objeto fue localizado y en qué posición geográfica. Con estos datos 40 cinemáticos de informaciones combinadas, tal como velocidad (speed), aceleración, etc., pueden proporcionarse también además de los datos de localización que comprenden, a modo de ejemplo, las coordenadas x, y y z. En la secuela de esta especificación, los datos cinemáticos y de localización proporcionados por el sistema de sensores de localización se definirán también como datos de sensores (sin tratar).

45 En una realización particular, a modo de ejemplo, de un sistema de seguimiento inalámbrico las personas u objetos pueden estar provistos de transmisores diminutos, que puedan integrarse en el calzado, uniformes y balones y cuyas señales se captan por varias antenas situadas alrededor del área objeto de observación. Las unidades receptoras procesan las señales recogidas y determinan sus valores de tiempo de llegada (ToA). Sobre la base de un cálculo de las diferencias en el retardo de propagación, la posición de cada transmisor se determina entonces de forma continua. Además, una red informática integrada con el sistema de seguimiento inalámbrico puede analizar los 50 datos de sensores o de posiciones con el fin de detectar eventos operativos específicos. Operando en la banda de 2.4 o 5 GHz, el sistema de seguimiento está globalmente libre de licencias.

Sobre la base de los flujos de datos de sensores, sin procesamiento, procedentes del sistema de seguimiento de 55 posiciones o localización así denominados "eventos" pueden detectarse. De este modo, un evento o tipo de evento puede definirse para ser una aparición instantánea de interés en un punto en el tiempo y puede definirse por un identificador ID de evento único. En general, un evento está asociado con un cambio en la distribución de una magnitud relacionada que se puede detectar. Una instancia de evento operativo es una ocurrencia instantánea de un tipo de evento en un punto en el tiempo distinto. Un evento puede ser un evento primitivo, que está directamente basado en datos de sensores (datos cinemáticos) del sistema de seguimiento, o un evento compuesto, que está 60 basado, en cambio, en otros eventos anteriormente detectados. Es decir, un evento compuesto no es directamente dependiente de los datos de sensores sin tratamiento sino de otros eventos. En aplicaciones de juegos de balón, un evento puede, a modo de ejemplo, ser "el jugador X golpea el balón" o "el jugador X está en posesión del balón". Eventos más complicados pueden ser, a modo de ejemplo, "fuera de juego" o "falta". Cada instancia de evento puede tener tres marcas temporales: una ocurrencia, una detección y una marca temporal derivada. Todas las marcas temporales están en el mismo dominio temporal discreto. La marca temporal de ocurrencia t_s es el tiempo 65

cuando el evento ha sucedido realmente, la marca temporal de detección dts es el tiempo cuando el evento ha sido detectado por un detector de eventos y la marca temporal de llegada ats es el tiempo cuando el evento fue recibido por un nodo particular de sistema de procesamiento de eventos (EPS). Las marcas temporales de ocurrencia y de detección están fijadas para una instancia de evento operativo en cualquier nodo receptor mientras que la marca temporal de llegada puede variar en diferentes nodos en la red.

La detección de eventos (Procesamiento de Eventos Complejos, CEP) basada en flujos de datos de sensores subyacentes ha suscitado un mayor interés en la base de datos y comunidades de sistemas distribuidos en los últimos años. Una amplia gama y números cada vez mayores de aplicaciones actualmente, incluyendo las aplicaciones como supervisión de red, empresa electrónica, cuidados sanitarios, análisis financiero y seguridad o la supervisión de eventos deportivos antes citados, se basan en la capacidad para procesar consultas sobre flujos de datos que adoptan, en condiciones ideales, la forma de series de eventos ordenados en el tiempo. La detección de eventos indica el procesamiento completamente automatizado de datos de sensores sin tratamiento y/o eventos sin la necesidad de intervención humana, puesto que en numerosas aplicaciones la gran cantidad de datos de sensores suministrados y/o eventos ya no pueden capturarse o procesarse por un operador humano. A modo de ejemplo, las publicaciones de documentación WO 2011/127059 A1 Endler M. et al “Métodos generales para realizar una transferencia sin discontinuidades” POMC’02. PROCEEDINGS OF THE 2ND. ACM TALLER INFORMÁTICO INTERNACIONAL SOBRE PRINCIPIOS DEL CÁLCULO MÓVIL. TOULOUSE, FRANCIA, 30-31 octubre 2002, están relacionados con la migración de eventos.

A modo de ejemplo, si están previstas variaciones de alta velocidad de jugadores o un objeto deportivo, p.ej., un balón, los datos de sensores sin procesamiento (seguimiento de posición o localización) han de determinarse en una tasa de datos suficientemente alta por la red de sensores (inalámbricos) subyacentes. Además, si existe un alto número de jugadores y/o objetos (p.ej., en un partido de fútbol existen 22 jugadores y un balón) a ser objeto de seguimiento mediante una cantidad de muestras de datos cinemáticos y de posiciones geográficas globales por segundo que puede hacerse prohibitivamente alta, en particular, con respecto a los requisitos de procesamiento en tiempo real.

En consecuencia, aun cuando flujos de datos de eventos y/o sensores sin procesamiento sean analizados y señalizados con un sistema completamente automatizado, puede ser todavía demasiada información, lo que posiblemente no tiene ningún interés en su integridad. En el futuro, este problema será cada vez más desfavorable puesto que cuantos más dispositivos estén provistos de sensores y la posibilidad de proporcionar sus datos de sensores determinados a las redes públicas tal como Internet para, (p.ej., datos de temperaturas o meteorológicos determinados por dispositivos inalámbricos tales como teléfonos inteligentes). Por este motivo, la cantidad de datos de sensores a procesarse en determinados eventos de interés crecerá con rapidez. La detección de eventos automatizada puede proporcionar remedio a esta situación intentando una agregación del elemento de datos de sensores, sin procesamiento, elemento por elemento, para determinar eventos más abstractos e interdependientes, que puedan transferir tanta más información que los propios datos de sensores sin procesamiento. A modo de ejemplo, además de los eventos relacionados con el fútbol antes citados, dichos eventos determinados podrían incluir “el vehículo X está situado en el cruce Y” o “congestión de tráfico en la ruta X”,

El problema que surge en la detección de eventos automatizada es la potencia de cálculo informático requerida para realizar la detección de eventos sobre flujos de datos de eventos y/o sensores en paralelo, en una cantidad posiblemente masiva – y todo ello bajo al menos requisitos de procesamiento casi en tiempo real. Este problema puede resolverse mediante la denominada paralelización de detectores de eventos, que puede, a modo de ejemplo, ejecutarse en nodos de redes diferentes (esto es, distribuidos) de una red informática, que pueden, a modo de ejemplo, comunicarse a través de Ethernet. De este modo, un detector de eventos extrae automáticamente un determinado evento de interés a partir de un flujo de datos de sensores o eventos en conformidad con las especificaciones de eventos de un usuario. Detectores de eventos individuales pueden distribuirse a través de nodos de redes diferentes de una red de datos, en donde los detectores de eventos diferentes se comunican utilizando datos de sensores y/o eventos que se desplazan a través de la red utilizando rutas y bifurcaciones de redes diferentes. De este modo, los datos de sensores sin procesamiento y/o eventos pueden transportarse en paquete de datos en conformidad con algún protocolo de transporte, tal como, a modo de ejemplo, UDP (Protocolo de Datagramas de Usuarios), TCP (Protocolo de Control de Transmisión)/IP (Protocolo Internet), etc. Este concepto, sin embargo, hace que se planeen nuevos problemas con respecto a una carga de cálculo informático posiblemente desequilibrada entre diferentes nodos de red y con respecto a la sincronización de flujos de datos de eventos dentro de la red. Sin contramedidas adecuadas, las cargas de cálculo informático entre diferentes nodos de redes están desequilibradas y los flujos de datos de eventos y/o sensores individuales en la red no están sincronizados temporalmente entre sí, lo que significa que los eventos individuales pueden alcanzar a un detector de eventos fuera de su orden temporal original y esta circunstancia, da lugar a resultados tetados falsos.

A modo de ejemplo, vamos a examinar el caso, a modo de ejemplo, de un escenario operativo de fútbol, en donde una pluralidad de detectores de eventos que funcionan automáticamente en paralelo se supone que detectan un pase desde el jugador A al jugador B. Con el fin de detectar el evento del “pase”, se requiere la siguiente secuencia de eventos precedente.

1. “el jugador A está en posesión del balón”
2. “el jugador A golpea el balón”
- 5 3. “el balón abandona al jugador A”
4. “el balón llega cerca del jugador B”
- 10 5. “el jugador B golpea el balón”.

La detección de eventos para el caso de “el jugador X golpea el balón” puede basarse en la secuencia de eventos de “el jugador X está cerca del balón” y un máximo de aceleración detectada del balón. Existen las siguientes alternativas para establecer un detector de eventos automatizado para dicho evento de “el jugador X golpea el balón”.

Podemos esperar eventos requeridos individuales – uno tras otro. Si hemos visto todos los eventos requeridos en el orden (temporal) correcto (en este caso, cualquier criterio de interrupción operativa se descarta para fines de una mayor simplicidad) podemos afirmar que hemos visto o ejecutado un pase. Sin embargo, para aplicaciones complejas, la detección de todos los eventos requeridos no tiene lugar necesariamente en un nodo de red único o una CPU (Unidad Central de Procesamiento) debido a la paralelización de detectores de eventos. Por este motivo, no está necesariamente garantizado que los eventos requeridos individuales alcancen al detector de eventos en el orden requerido correcto. Esta circunstancia puede, a modo de ejemplo, deberse a fluctuaciones operativas de las redes, carga de la unidad CPU variable y/o desequilibrada o aumento de la carga de la red. A modo de ejemplo, consideremos un flujo de eventos constituido por instancias de eventos operativos e_1, e_2, \dots, e_n , con $e_k.ats < e_{k+1}.ats$ ($1 \leq k < n$), esto es, los eventos en el flujo de eventos se clasifican por su hora de llegada en orden ascendente. Si cualquier evento e_i con e_j con $1 \leq i < j \leq n$, existe, en tal caso $e_i.ts > e_j.ts$ el evento e_j se indica como un evento fuera de orden.

En consecuencia, podríamos intentar memorizar los eventos y luego, buscar en la memoria el modelo de eventos correcto. ¿Pero qué magnitud de memoria debe utilizarse?. Si afirmamos que se ha producido un pase dentro de un máximo de 5 segundos, tendríamos que considerar eventos dentro de un periodo de tiempo de un máximo de 5 segundos después de que el primer evento pertinente hasta que hayamos detectado el pase o hasta que lo abortemos operativamente. Sin embargo, es también posible que el último evento pertinente sea de cálculo bastante complejo, lo que requiere una pequeña memoria intermedia adicional. ¿Pero cuál es la magnitud de esta memoria intermedia adicional? ¿Y cuál es la capacidad de la memoria en relación con los detectores de eventos compuestos que requieren el evento de pase como un evento de entrada?

El algoritmo de K-slack de S. Babu, U. Srivastava y J. Widom, “Explotación de k-demands para reducir la sobrecarga de memoria en consultas continuas sobre flujos de datos”, ACM Trans. Database Systems, vol. 29, páginas 545-580, 2004, es una solución bien conocida para el tratamiento de los eventos fuera de orden en la detección de eventos operativos. El algoritmo K-slack utiliza una memoria intermedia de longitud K para cerciorarse de que un evento operativo e_i , puede retardarse en como máximo, K unidades de tiempo (K ha de ser conocido *a priori*). Sin embargo, en un sistema distribuido, los retardos de señalización de eventos dependen de una configuración del sistema/red completa, esto es, la distribución de los detectores de eventos, así como la carga de la red y de la CPU. Ni la configuración del sistema final ni el escenario operativo de carga pueden ser previstos en el momento de la compilación.

Un método establecido por M. Li, M. Liu, L. Ding, E. A. Rundensteiner y M. Manipulación, “Procesamiento de flujos de eventos con llegada de datos fuera de orden”, en Proc. 27th Intl. Conf. Talleres informativos de sistemas informáticos distribuidos (Washington, DC), páginas 67-74, 2007, memoriza un evento e_i , al menos tan largo como $e_i.ts + K \leq clk$. Puesto que no existe ningún reloj global en un sistema distribuido, cada nodo sincroniza su reloj local ajustándolo a la marca temporal de ocurrencia de mayor magnitud vista hasta ahora.

Una unidad de ordenamiento que pone en práctica el método del algoritmo de K-slack aplica una ventana deslizante con un valor de K dado para el flujo de entrada, retarda los eventos en conformidad con sus marcas temporales y genera un flujo de salida ordenado de eventos. Sin embargo, un valor K unido, fijo *a priori*, no sirve para detectores de eventos jerárquicos distribuidos. Puesto que el algoritmo de K-slack toma K unidades de tiempo para generar un evento compuesto, un detector de eventos en una capa superior que memoriza también K unidades y espera al evento compuesto, pierde dicho evento. Los tiempos de espera se suman a lo largo de la jerarquía de detectores de eventos.

En el documento de M. Liu, M. Li, D. Golovnya, E. Rundensteiner y K. Claypool, titulado “Procesamiento de consultas de modelos de secuencias de flujos de eventos fuera de orden”, en Proc. 25th Intl. Conf. Ingeniería de datos (Shanghái, China), páginas 784-795, 2009, se evitan dichos problemas especificando un valor de K individual para cada detector de eventos. Cada K_n (indicando n el nivel de jerarquía) debe establecerse para un valor mayor que $\max(K_{n-1})$, esto es, mayor que el retardo de máximo de todos los eventos suscritos. De este modo, un evento

suscrito es un evento de interés para el detector de eventos respectivo. El detector de eventos de nivel de jerarquía n se suscribe a un evento de un nivel de jerarquía inferior con el fin de utilizarlo como una entrada para detectar un evento de jerarquía más alta. Aunque lo que antecede parece adecuado en primera instancia, resulta difícil elegir valores adecuados para todos los parámetros K_j , que son específicos de la aplicación y de la topología, y sólo pueden realizarse después de medidas rigurosas. Desde un punto de vista conservativo y duradero, los grandes valores de K_j dan lugar a la necesidad de grandes memorias intermedias para satisfacer las altas demandas de memoria y dando lugar a largos retardos para CEP jerárquica (puesto que se añaden los retardos). Debe evitarse, por lo tanto, valores de K_j demasiado grandes. En teoría, para un sistema de uso general el más pequeño/mejor valor de K_j solo puede encontrarse por medio de mediciones en el tiempo de ejecución puesto que las latencias dependen de la distribución de detectores de eventos y de la topología de red subyacente completa. Además, los mejores valores de K_j cambian en el tiempo de ejecución cuando los detectores realizan una migración operativa.

Con un flujo de eventos entrantes e_i , una idea clave para superar los problemas antes citados es realizar estas mediciones de tiempo de ejecución comparando una marca temporal de ocurrencia de un evento t_s con su marca temporal de llegada t_{at} . Es decir, las formas de realización de la presente invención están basadas en el supuesto de que una recuperación de un orden temporal original de eventos que alcanza a un detector de eventos por intermedio de rutas de la red y , por lo tanto, experimentando diferentes retardos de procesamiento y/o propagación $\delta(\cdot)$, puede conseguirse retardando los eventos de forma adecuada antes de su reenvío o retransmisión a un detector de eventos posterior. El tiempo en el que se retransmite un evento para el detector de eventos (de flujo descendente) posterior puede basarse en la marca temporal original del respectivo evento y los retardos de procesamiento y/o propagación de todos los eventos de entrada que se requieren por el detector de eventos posterior con el fin de determinar su evento de salida.

Con el fin de garantizar un valor de retado común requerido mínimo, esto es, para permitir una retransmisión posiblemente rápida de los (al menos) dos eventos, un tiempo de salida para la retransmisión de un primero y un segundo eventos de entrada a su detector de eventos en flujo descendente asociado puede determinarse sobre la base del primero y del segundo valor de temporización de eventos (del evento respectivo) y sobre la base de un retado máximo del primero y del segundo eventos. En consecuencia, un retado $\delta(\cdot)$ de un evento puede deberse a varios motivos. A modo de ejemplo, un retado de evento dentro del sistema informático distribuido puede deberse a diferentes condiciones de fluctuaciones en diferentes rutas de la red o deberse a diferentes duraciones de procesamiento y/o latencias de red diferentes. Dicho de otro modo, un orden de recepción del primero y del segundo evento a la entrada del compensador de retardos puede ser diferente de un orden de ocurrencia original del primero y del segundo eventos debido a diferentes condiciones de fluctuación, diferentes duraciones del procesamiento y/o diferentes latencias de la red.

Los retardos de eventos del primero y del segundo eventos, respectivamente, pueden medirse o determinarse sobre la base de un tiempo de recepción/llegada del primero y del segundo en una unidad de ordenamiento y sobre la base de sus valores de temporización de eventos asociados respectivos que reflejan las ocurrencias de los eventos. Cuando se indica un retado de propagación o de señalización de un evento e_i ($i = 1, 2, \dots$) desde su tiempo de ocurrencia o detección $t_{event,ei}$ ($i = 1, 2, \dots$) a la entrada del compensador de retardos mediante $\delta(\cdot)$, la unidad de ordenamiento puede ser utilizable para determinar las instancias de tiempos de salida $t_{out,ei}$ ($i = 1, 2, \dots$) para retransmitir el primero y el segundo eventos para el detector de eventos posterior sobre la base de

$$t_{out,e1,e2} = t_{event,e1,e2} + \max(\delta(e_1), \delta(e_2)).$$

en donde $\max(\cdot)$ indica el operador máximo. Es decir, la unidad de ordenamiento puede ser utilizable para determinar un valor de retado máximo tomado a partir del conjunto de retardos de eventos asociados con el primero y el segundo eventos. En el caso de que existan más de dos eventos a retardarse por la unidad de ordenamiento, en tal caso, por supuesto, la unidad de ordenamiento puede determinar el valor de retado común tomando el valor de retado máximo desde el conjunto de retardos de eventos asociados para los más de dos eventos.

Debido a diversos efectos dentro del sistema distribuido, el valor para $K = \max(\delta(e_1), \delta(e_2))$ puede cambiar. Además, cuando clk cambia de forma imprevista en un detector de eventos, esto puede dar lugar a supuestos erróneos para K y, por lo tanto, a salidas de detectores incorrectas.

El primer problema antes citado, esto es, un aumento imprevisto de K_j es raro puesto que, con frecuencia, los retardos de eventos permanecen estables. Si se produce, no podemos garantizar un ordenamiento correcto en general. No obstante, un margen de seguridad probabilística para retardos pueden proporcionarse para atenuar este problema. Por el contrario, siempre podemos evitar el segundo problema, esto es, cambios imprevistos de clk . Unas formas de realización de la idea inventiva de la presente invención pueden basarse en no ajustar de forma más prolongada el reloj para la mayor marca temporal vista hasta ahora sobre cualesquiera eventos entrantes, sino solamente utilizar un tipo designado de evento para el ajuste de clk .

Un cambio imprevisto de clk puede evitarse si ajustamos un reloj del nodo de red solamente en algunos tipos de

eventos entrantes. Aunque la definición anterior de eventos fuera de orden ha identificado eventos que son tardíos, utilizamos ahora los valores de clk para posponer eventos que llegan demasiado pronto. Entre cualesquiera dos actualizaciones de clk , los eventos no pospuestos se ordenan en conformidad con sus marcas temporales. Más concretamente, consideremos un flujo de eventos e_1, e_2, \dots, e_n como antes. Supongamos que clk solamente se establece por eventos de tipo ID. Un evento e_j es un evento fuera de orden, si no existe ningún e_i, e_k con $e_i.ts = e_k.ts = ID$ de modo que $e_i.ts \leq e_j.ts \leq e_k.ts$. El retado de eventos de e_i puede darse por $\delta(e_i) = e_k.ts - e_j.ts$

La cuestión restante es qué tipo de evento captar para ajustar clk . Cuanto mayor sea la frecuencia de ocurrencia del tipo de evento captado, tantos menos eventos necesitan posponerse, tantos más pequeños son los valores de K resultantes y tanto mejor son los retardos medidos. Si existe una opción de elegir tipos de eventos, es preferible el que tiene el retado más estable y fijo, puesto que se refleja mejor en el tiempo real. Si por el contrario, los eventos de un cierto tipo varían en sus tiempos de llegada, clk no se comporta sin discontinuidades operativas. Para aumentar la frecuencia de actualización del reloj, en lugar de utilizar solamente un tipo de evento, es posible utilizar un conjunto de tipos de eventos para ajustar el reloj, a condición de que dichos tipos de eventos tengan los mismos retardos absolutos, a modo de ejemplo, si son eventos de sensores desde la misma fuente. Los eventos de sensores de alta tasa de datos, con marcas temporales precisas, son candidatos excelentes.

Tomando los eventos de sensores, esto es, datos de sensores sin procesamiento, como eventos de actualización de reloj estables, nos permite determinar un valor adecuado para K . No obstante, el primer problema de aumentos imprevistos de K está todavía sin resolver. Si K es un valor demasiado pequeño, omitimos eventos o su proceso fuera de orden, y en tal caso, aumentamos el valor de K para resolver la presencia de retardos de eventos futuros. Con un margen de seguridad añadido, esto es, un valor de K ligeramente mayor, pueden evitarse dichos errores de detección. En lugar de fijar K después de que haya ocurrido un error, es posible ajustar K teniendo en cuenta la variación prevista de los retardos. Es preciso recordar que K se define como el retado máximo de todos los eventos suscritos de un detector de eventos. La estimación de los retardos de los eventos particulares da lugar más exactamente a un valor de K más seguro. Para tener un mejor supuesto de retado que esperamos para un evento e_i , es posible utilizar todas las mediciones de retardos recientes de e_i y determinar su desviación estándar. El retado supuesto es, entonces, el retado máximo de e_i más el producto de la desviación estándar y un factor de escalamiento λ . El factor λ puede definirse por el arquitecto del sistema y tiene influencia sobre la probabilidad de que K será suficientemente grande. Con dichos valores de K , podemos ordenar los flujos de eventos de entrada incluso con retardos de eventos inestables.

Sin embargo, si un valor de K crece para un determinado detector de eventos, este hecho permanece desconocido para una suscripción de detector de eventos superior a la jerarquía del detector. El detector de nivel superior solamente notificará un retado demasiado grande, de forma potencial y variante, cuando el evento suscrito sea realmente generado. En tal caso, el K de nivel superior puede ser demasiado pequeño para evitar la falta de detección y la regeneración de K . No obstante, el detector de nivel superior podría haber modificado su valor de K con prontitud suficiente si solamente se hubiera informado con anterioridad. Por lo tanto, cuando K cambia es posible notificar a los abonados de modo que puedan modificar sus valores de K si fuere necesario. Para ello, es posible enviar inmediatamente un pseudo-evento con una marca temporal adecuada. El destinatario solamente puede utilizar dicho pseudo-evento para fines de configuración.

Ahora, disponemos de un valor de K suficientemente adecuado y un reloj estable clk . Sin embargo, esta circunstancia no proporciona todavía un flujo de eventos ordenado o clasificado. Necesitamos una unidad de ordenamiento de eventos que funcione adecuadamente sobre un flujo fuera de orden para proporcionar una entrada clasificada al detector posterior. Dicha unidad de ordenamiento puede ponerse en práctica como una denominada 'caja negra' y puede solamente montarse entre el flujo de eventos original y la entrada del detector de eventos, de modo que no exista ninguna necesidad de modificar el propio detector de eventos. El flujo de salida de la unidad de ordenamiento es una secuencia clasificada de eventos con un retado mínimo. Siempre que se reciba un evento de entrada (siendo ignorado los pseudo-eventos) se clasifican en una memoria intermedia de la unidad de ordenamiento en conformidad con su marca temporal de ocurrencia. Si los eventos fuera de orden son raros, la inserción de la clasificación de nuevos eventos normalmente es tan sencilla y rápida que se lleva a la cabecera de la memoria intermedia. Siempre que se actualiza clk y $e_i.ts + K \leq clk$ se mantiene para algunos eventos de salida e_i en la memoria intermedia, dichos e_i pueden emitirse al flujo de salida de la unidad de ordenamiento y se puede procesar el siguiente evento de entrada.

Conviene señalar que normalmente existe más de un detector de eventos por máquina o nodo de red, cada uno de los cuales puede tener una unidad de ordenamiento dedicada con un valor de K específico del detector que solamente captación sus eventos suscritos desde el flujo de eventos principal.

En el procesamiento de eventos distribuidos puede ser necesario desplazar un detector de eventos desde una máquina a otra en el tiempo de ejecución. Esta circunstancia puede causarse por numerosos motivos, p.ej., las máquinas pueden necesitar una parada técnica para fines de mantenimiento o debido a fallos de sistemas o las máquinas pueden tener una sobrecarga o estar incluso agotadas y no pueden realizar sus tareas de procesamiento de eventos con la suficiente rapidez. Dichos cambios pueden ser un resultado de la dinámica en el entorno analizado, a modo de ejemplo por disponer de existencias no previstas.

El trabajo relacionado sobre la migración en el tiempo de ejecución de procesos u objetos se realiza principalmente en el área de las máquinas virtuales (VMs). A modo de ejemplo, en el documento de R. Bradford, E. Kotsovinos, A. Feldmann y H. Schioberg, titulado "Migración directa en área amplia de máquinas virtuales incluyendo un estado persistente local", en Proc. 3rd Intl. Conf. Entornos de ejecución virtual (San Diego, CA) páginas 169-179, 2007, se refiere a la transferencia del estado de VM persistente local. Después de la migración, las conexiones de red son redireccionadas para el nuevo host y las órdenes de las antiguas conexiones son objeto de reenvío. La antigua VM se cierra tan pronto como se concluyen las conexiones antiguas. Sin embargo, ambas máquinas no solamente tienen que ejecutarse en paralelo mientras se están reenviando órdenes, pero se ignora el orden en el que se reciben órdenes a través de la red.

El documento de CR/TR-Motion H. Liu, H. Jin, X. Liao, L. Hu y C. Yu titulado "Migración directa de máquina virtual sobre la base de la traza y reproducción completa del sistema" en Proc. 18th ACM Intl. Symp. Sistema informático distribuido de alto rendimiento" (Garching, Alemania), páginas 101-110, 2009 utiliza la tecnología de punto de control/recuperación y traza/reproducción para conseguir una migración rápida de VMs. Los puntos de control desde la VM origen se recuperan en el destino y las trazas de llamadas desde el origen se retransmiten de modo que ambas máquinas sean compatibles. El tiempo inactivo es notablemente inferior al de los métodos anteriores. Sin embargo, los autores no consideran que el orden de las órdenes entrantes pueda ser diferente en el nuevo host.

El método establecido en el documento de V. Medina y J. M. García, "Replicación en directo de máquinas virtuales" en Proc. 10th WSEAS Intl. Conf. Sistemas distribuidos y en paralelo, en Ingeniería de software (Stevens Point, WI), páginas 15-23, 2011, es similar al de CR/TR-Motion pero asume réplicas directamente desde el principio. El usuario interactúa con una sola máquina virtual VM pero se envían también órdenes a una réplica de modo que ambas máquinas virtuales VMs estén siempre sincronizadas. Cuando la máquina antigua deja de trabajar, el protocolo del lado del cliente envía las órdenes a la VM replicada y presenta la respuesta de la réplica. Sin embargo, las órdenes son redireccionadas desde el lado de los clientes, lo que significa que el ordenamiento puede ser distinto después de la conmutación a la réplica.

Ninguno de los métodos considera el orden de las órdenes entrantes y/o datos de forma explícita. Ello se debe a que normalmente el origen de las órdenes, esto es, la estación de trabajo del usuario, es estática y las órdenes se siguen recibiendo en el orden correcto. Sin embargo, si se trata de máquinas virtuales VMs multiusuario, pueden surgir problemas si dos usuarios intentan modificar el mismo fichero. En la máquina VM original, la orden del usuario A puede recibirse en primer lugar, mientras que en la máquina VM migrada, la orden del usuario B estará en primer lugar. Las máquinas virtuales VMs están entonces fuera de sincronización. Métodos tales como CR/TR-Motion repetirían el proceso de recuperación y reproducción en dichas improbables situaciones. Sin embargo, para la detección de eventos operativos, dichas situaciones son muy probables y la migración nunca terminaría.

En consecuencia, es deseable mejorar la tecnología aplicada para realizar una migración eficiente y segura de detectores de eventos que operan sobre eventos fuera de orden entrantes desde un nodo de red a otro.

SUMARIO DE LA INVENCIÓN

Para detectar eventos operativos, tales como eventos primitivos y/o compuestos, varios detectores de eventos pueden estar funcionando en diferentes nodos de un sistema informático distribuido. Dichos detectores de eventos pueden causar diferentes cargas sobre sus respectivos nodos asociados y/o el sistema informático distribuido completo, dependiendo de la condición del sistema y/o dependiendo de la complejidad del cálculo informático del algoritmo subyacente del detector de eventos y/o los datos de sensores objeto de análisis. Por consiguiente, en un escenario operativo de instalaciones o condiciones variables de una configuración del sistema original, una distribución original de los detectores de eventos en los diversos nodos de red podría resultar sub-óptima con respecto al equilibrio de la carga de cálculo informático o podría incluso dar lugar a un desplome del sistema informático distribuido.

Dichos desequilibrios de carga indeseables pueden superarse mediante formas de realización de la presente invención. Si, a modo de ejemplo, al menos un retardo de eventos alcanza un valor prohibitivamente grande, puede constituir u indicador para ese hecho en un detector de eventos origen, desde donde el evento correspondiente alcanza un detector de eventos destino, que esté sobrecargado o experimente algunos otros problemas operativos. En dicho ámbito, puede ser deseable transferir o reubicar el proceso u objeto del detector de eventos origen defectuoso a algún otro recurso de hardware del sistema informático distribuido. A modo de ejemplo, el proceso u objeto de detector de eventos origen defectuoso puede transferirse o ser objeto de migración desde su nodo actual a otro nodo diferente y físicamente separado del sistema informático distribuido, que tiene más recursos de hardware disponibles.

Es un hallazgo de la presente invención que la migración de un detector de eventos desde una máquina a otra máquina da lugar a diferentes retardos de eventos de los eventos gestionados por el detector de eventos afectado. Cuando un detector de eventos es objeto de migración desde un primer nodo de red a un segundo nodo de red, los nodos origen de los eventos suscritos permanecen siendo los mismos, pero los retardos de eventos relacionados

5 cambiarán por supuesto. A no ser que los retardos se reduzcan para todos los eventos implicados, es probable que falle una migración del detector de eventos sin procesamiento debido a que el detector de eventos migrados (o cualesquiera detectores de eventos en una capa superior) ya no detecten los eventos suscritos en un orden correcto porque el valor de K que era adecuado en el primer nodo (antiguo) es demasiado pequeño para el segundo (nuevo) nodo y los nuevos retardos de eventos resultantes.

Por consiguiente, es un objetivo de la presente invención garantizar eventos de entrada ordenados para el detector de eventos objeto de migración.

10 Si un hallazgo de la presente invención es que para una transferencia cooperativa del detector de eventos el nuevo nodo de un sistema informático distribuido no solamente suscribe eventos de entrada necesarios desde sus nodos origen o fuente, pero el nodo antiguo reenvía entonces dichos eventos también al nuevo nodo. El nuevo nodo puede derivar el orden de eventos correcto combinando la información de retardos de eventos que llegan en el nuevo nodo a lo largo de dos rutas, esto es, la ruta reenviada (desde los nodos origen por intermedio del nodo antiguo) y la ruta directa (desde los nodos origen directamente).

15 En conformidad con un primer aspecto de la idea inventiva, se da a conocer un método para efectuar la migración de un detector de eventos (proceso u objeto) desde un primer nodo (esto es, el nodo antiguo) de un sistema informático distribuido a un segundo nodo (esto es, nuevo nodo) del sistema informático distribuido. El detector de eventos procesa una pluralidad de eventos operativos de entrada para generar al menos un evento de salida sobre la base de los eventos de entrada. Cada uno de los eventos de entrada está asociado con un retardo de eventos correspondiente a un tiempo de propagación o retardo del respectivo evento de entrada para el primer nodo, en donde el tiempo de propagación puede considerarse como una diferencia entre un tiempo de ocurrencia de evento y una recepción de evento posterior o el tiempo de llegada del respectivo evento en el primer nodo. El método comprende una etapa de transferir, desde el primer nodo al segundo nodo, información sobre los retardos de eventos correspondientes a la pluralidad de eventos de entrada en el primer nodo, una etapa de transferencia de un contenido (proceso u objeto) del detector de eventos desde el primer nodo al segundo nodo para obtener un detector de eventos objeto de migración en el segundo nodo, una etapa de reenviar la pluralidad de eventos de entrada desde el primer nodo al segundo nodo, una etapa de recepción en el segundo nodo, de la pluralidad de eventos de entrada reenviados desde el primer nodo y, en paralelo, desde los detectores de eventos origen asociados con la pluralidad de eventos de entrada, una etapa de procesamiento (ordenado), en el segundo nodo, de la pluralidad de eventos de entrada recibidos sobre la base de la información transferida sobre los retardos de eventos y sobre la base de los retardos de eventos de la pluralidad de eventos de entrada recibidos desde los detectores de eventos asociados.

20 25 30 35 En conformidad con otro aspecto de la idea inventiva, se da a conocer también un sistema informático distribuido para la migración de un detector de eventos desde un primer nodo del sistema informático distribuido a un segundo nodo del sistema informático distribuido. El sistema informático distribuido comprende el primer nodo que tiene medios para transferir, desde el primer nodo al segundo nodo, información sobre los retardos de eventos correspondientes a la pluralidad de eventos de entrada en el primer nodo, medios para transferir un contenido del detector de eventos desde el primer nodo al segundo nodo para obtener un detector de eventos objeto de migración en el segundo nodo, y medios para reenviar la pluralidad de eventos de entrada desde el primer nodo al segundo nodo. El sistema informático distribuido comprende, además, el segundo nodo que tiene medios para la recepción, en el segundo nodo, de la pluralidad de eventos de entrada reenviados desde el primer nodo y, en paralelo, desde los detectores de eventos de origen asociados a la pluralidad de eventos de entrada, medios para el procesamiento, en el segundo nodo, de la pluralidad de eventos de entrada recibidos sobre la base de la información transferida sobre los retardos de eventos y sobre la base de los retardos de eventos de la pluralidad de eventos de entrada recibidos desde los detectores de eventos asociados.

40 45 50 En consecuencia, las formas de realización dan a conocer también el primero y/o el segundo nodo del sistema informático distribuido.

55 Por lo tanto, también se da a conocer un primer método (para el primer nodo) para la migración de un detector de eventos desde el primer nodo al segundo nodo del sistema informático distribuido. El método comprende una etapa de transferir, desde el primer nodo al segundo nodo, información sobre los retardos de eventos que corresponden a la pluralidad de eventos de entrada en el primer nodo, una etapa de transferencia de un contenido o estado del detector de eventos desde el primer nodo al segundo nodo para obtener un detector de eventos objeto de migración en el segundo nodo y una etapa de reenviar la pluralidad de eventos de entrada desde el primer nodo al segundo nodo.

60 65 Asimismo, se da a conocer un segundo método (del segundo nodo) para efectuar la migración de un detector de eventos desde el primer nodo al segundo nodo del sistema informático distribuido que comprende una etapa de recepción, en el segundo nodo, de la pluralidad de eventos de entrada reenviados desde el primer nodo y, en paralelo, desde detectores de eventos origen asociados a la pluralidad de eventos de entrada y una etapa de procesamiento, en el segundo nodo, de la pluralidad de eventos de entrada recibidos sobre la base de la información transferida sobre los retardos de evento y sobre la base de retardos de eventos de la pluralidad de eventos de

entrada recibidos desde los detectores de eventos asociados.

Algunas formas de realización comprenden circuitos de control digital instalados dentro de los nodos o aparatos. Dichos circuitos de control digital, a modo de ejemplo, un procesador de señal digital (DSP) o un circuito integrado específico de la aplicación (ASIC), necesita programarse en consecuencia. Por lo tanto, todavía otras formas de realización dan a conocer también un programa informático que tiene un código de programa para realizar el método de la presente invención, cuando el programa informático se ejecuta en un ordenador o en un procesador digital.

En conformidad con las formas de realización, un detector de eventos ha de entenderse como una instancia de un programa informático que se está ejecutando un nodo del sistema distribuido. Un detector de eventos comprende el código de programa del programa informático y su actividad en curso. El sistema distribuido puede ser una red informática distribuida o un procesador de núcleos múltiples, a modo de ejemplo. En caso de una red informática, un nodo, esto es, un nodo de red, puede comprender un dispositivo informático o una unidad de procesamiento (p.ej., CPU) que se comunican con otros nodos de red por intermedio de Ethernet, a modo de ejemplo, o alguna otra forma de tecnología de gestión de redes. Es decir, en conformidad con otro aspecto de la idea inventiva de la presente invención, se da a conocer también un sistema informático distribuido para determinar eventos basados en al menos un flujo de datos de sensores (sin procesamiento). El sistema informático distribuido dado a conocer, que puede ser una red informática, comprende una pluralidad de nodos distribuidos, teniendo cada uno de ellos un detector de eventos asociado y al menos una forma de realización de un aparato para efectuar la migración de detectores de eventos entre los nodos distribuidos del sistema informático distribuido.

El procedimiento de migración o de transferencia de procesos del detector de eventos puede realizarse, a modo de ejemplo, con la ayuda de un Sistema Operativo (OS) que controla el sistema informático distribuido. Un sistema operativo OS ha de entenderse como un conjunto de programas que gestionan recursos de hardware de ordenador del sistema distribuido y proporciona servicios comunes para el software de aplicación, tal como los procesos u objetos del detector de eventos, a modo de ejemplo. Dichos servicios de sistemas operativos OS pueden comprender también funcionalidades de instanciación, esto es, funcionalidades para crear una instancia de proceso y/o de objeto para la que un aparato para migración puede tener acceso. Es decir, el aparato para efectuar la migración o el aparato de migración puede comprender medios para acceder a las funcionalidades de instanciación de procesos y/o objeto de un sistema operativo que controla el sistema distribuido.

En algunas formas de realización, el sistema informático distribuido puede estar acoplado a un sistema de localización para localizar y/o efectuar el seguimiento de objetos dentro de una zona geográfica predefinida, en donde el sistema de localización proporciona el por lo menos un flujo de datos de sensores al sistema informático distribuido, incluyendo el flujo de datos de sensores algunos datos que son indicativos de las posiciones geográficas de los objetos localizados. El sistema de localización puede estar basado en una red de sensores inalámbrica, que ha sido ya descrita en la parte introductoria de esta especificación de la memoria.

Un detector de eventos puede comprender una máquina de estados, que puede entenderse como un modelo de comportamiento utilizado para diseñar un programa informático subyacente de un proceso de detector de eventos. Una máquina de estados está constituida por un número (finito) de estados asociados con las transiciones, en donde la transición es un conjunto de acciones que se inicia desde un estado y finaliza en otro (o el mismo) estado. Una transición se inicia mediante una iniciación operativa, en donde dicha iniciación operativa puede ser, a modo de ejemplo, un evento primitivo o compuesto o una entrada de datos de sensores, sin procesamiento, hacia el detector de eventos. Por lo tanto, en conformidad con los aspectos de la idea inventiva de la presente invención, un detector de eventos puede comprender una máquina de estado y un contenido de proceso de detector de eventos o memoria que puede reflejar un estado actual de dicha máquina de estado tal como, a modo de ejemplo, variables individuales o conjuntos de variables. Para los procesos del detector de eventos, la máquina de estados puede ser una máquina de estados basada en software, p.ej., una máquina de estados de Lenguaje de Modelado Unificado (UML).

Una ventaja operativa de las formas de realización puede considerarse en la auto-organización de un sistema informático distribuido en conformidad con las formas de realización de la presente invención. Dicho sistema informático distribuido puede reaccionar ante el cambio de las condiciones del sistema efectuando una transferencia de los detectores de eventos entre nodos de red, de modo que el sistema distribuido o la red sea siempre capaz de realizar un procesamiento de señales de eventos eficiente y operativamente robusto.

En condiciones normales, no es posible hacer funcionar en paralelo el detector de eventos antiguo y el detector de eventos nuevo si la sobrecarga de la CPU o el desbordamiento de la capacidad de la memoria intermedia han iniciado operativamente la migración. Además, el reenvío de flujos de datos completo desde el nodo antiguo a nodo nuevo es prohibitivo puesto que pueda causar altas cargas de la red y sobrecargas de procesamiento durante un largo periodo de tiempo si eventos particulares se producen pocas veces. Las formas de realización proponen un concepto que permite la migración de detectores de eventos en el tiempo de ejecución e inicializa los valores de K en conformidad con los retardos de temporización en el nuevo nodo. La latencia introducida carece de importancia y ambas instancias de los detectores antiguos y nuevos así como detectores de nivel superior detectan eventos en orden en cualquier momento. El detector de eventos antiguo puede interrumpir su funcionamiento tan pronto como se haya copiado el estado y la sobrecarga de la gestión de redes se mantiene en un mínimo.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

5 Algunas formas de realización de aparatos y/o métodos se describirá a continuación a modo de ejemplo solamente y con referencia a los dibujos adjuntos, en donde

10 La Figura 1 ilustra, de forma esquemática, un diagrama de flujo de un método para la migración de un detector de eventos que funciona en un primer nodo a un segundo nodo de un sistema informático distribuido, en conformidad con una forma de realización,

15 Las Figuras 2a,b ilustran configuraciones, a modo de ejemplo, antes y después de la migración del detector de eventos,

La Figura 3 ilustra un diagrama de secuencias de mensajes para una migración de detector de eventos en conformidad con una forma de realización de la invención,

La Figura 4 ilustra retardos de eventos antes, durante y después de la migración del detector de eventos; y

20 La Figura 5 ilustra un pseudocódigo, a modo de ejemplo, para un programa informático para adaptación de los retardos y cancelación de los ecos operativos, en conformidad con una forma de realización de la invención.

DESCRIPCIÓN DETALLADA DE LAS FORMAS DE REALIZACIÓN

25 Varias formas de realización, a modo de ejemplo, se describirán a continuación con más detalle haciendo referencia a los dibujos adjuntos en los que algunas formas de realización, a modo de ejemplo, se ilustran a este respecto. En las Figuras, los espesores de las capas y/o zonas pueden estar exagerados en sus dimensiones para mayor claridad.

30 En consecuencia, aunque las formas de realización, a modo de ejemplo, son capaces de varias modificaciones y formas alternativas, sus formas de realización se ilustran, a modo de ejemplo, en las Figuras y se describirán aquí en detalle. Debe entenderse, sin embargo, que no existe ninguna intención de limitar las formas de realización, a modo de ejemplo, a las formas particulares aquí dadas a conocer sino que por el contrario, las formas de realización a modo de ejemplo son para cubrir todas las modificaciones, equivalentes y alternativas que caen dentro del alcance de protección de la invención. Los números similares se refieren a elementos idénticos o similares a través de toda la descripción de las Figuras.

35 Se entenderá que cuando un elemento se refiere como estando "conectado" o "acoplado" a otro elemento, puede conectarse directamente o acoplarse al otro elemento o a elementos intervinientes que puedan estar presentes. Por el contrario, cuando un elemento se refiere como estando "directamente conectado" o "directamente acoplado" a otro elemento, no existen elementos intervinientes presentes. Otros términos que se utilizan para describir la relación entre elementos deben interpretarse de una forma similar (p.ej., "entre" en comparación con "directamente entre", "adyacente" frente a "directamente, adyacente", etc.).

40 La terminología aquí utilizada es para la finalidad de describir formas de realización particulares solamente y no está prevista para ser limitadora de las formas de realización a modo de ejemplo. Tal como aquí se utilizan las formas singulares de "un", "una" y "el" o "la" están previstos para incluir también las formas plurales, a no ser que el contexto indique claramente lo contrario. Se entenderá, además, que los términos "comprende", "que comprende", "incluye" y/o "incluyendo", cuando aquí se utilizan, especifican la presencia de características operativas, números enteros, etapas, operaciones, elementos y/o componentes establecidos, pero no excluyen la presencia o adición de una o más de otras características, números enteros, etapas, operaciones, elementos, componentes y/o sus grupos.

45 A no ser que se defina de otro modo, todos los términos (incluyendo términos técnicos y científicos) aquí utilizados tienen el mismo significado que el que se suele entender por un experto en esta técnica a la que pertenecen las formas de realización a modo de ejemplo. Se entenderá, además, que los términos, p.ej., los definidos en diccionarios normalmente utilizados deben interpretarse como teniendo un significado que sea compatible con su significado en el contexto de la técnica pertinente y no habrán de interpretarse en un sentido idealizado o formal a no ser que se defina aquí expresamente de este modo.

50 La Figura 1 ilustra, de forma esquemática, un diagrama de flujo de un método 100 para efectuar la migración o transferencia de un detector de eventos que funciona inicialmente en un primer (antiguo) nodo o máquina sin discontinuidades para un segundo (nuevo) nodo o máquina de un sistema informático distribuido, en conformidad con una forma de realización de la presente invención.

55 El método de migración 100 comprende dos sub-métodos 110 y 120. Las etapas del sub-método 110 pueden asociarse con el primer nodo de red, en donde las etapas del segundo sub-método 120 pueden asociarse al segundo nodo de red, en conformidad con algunas formas de realización. Sin embargo, lo que antecede no implica

que los sub-métodos 110 y 120 solamente se realicen por el primero y el segundo nodo de red, respectivamente. En cambio, el método 110 puede realizarse en parte en el segundo nodo de red, a modo de ejemplo. Según se explicó con anterioridad, el primero y el segundo nodo del sistema informático distribuido, pueden ser, a modo de ejemplo, unidades CPU separadas. Dichas unidades CPU separadas pueden estar localizadas en dispositivos informáticos físicamente diferentes o pueden corresponder a diferentes núcleos de procesamiento independientes de una CPU de núcleos múltiples, a modo de ejemplo.

La instancia del detector de eventos objeto de migración desde el primer nodo al segundo nodo procesa una pluralidad de eventos de entrada suscritos con el fin de generar al menos un evento de salida compuesto, sobre la base de los eventos de entrada suscritos. Es decir, los eventos de entrada son de una jerarquía de evento inferior a la del evento de salida compuesto generado. Cada uno de los eventos de entrada suscritos está asociado con un retardo de evento individual $\delta(\cdot)$ que corresponde a una diferencia temporal entre una ocurrencia de eventos o un tiempo de detección (ts o dts) y una recepción de evento posterior o un tiempo de llegada (ats) del respectivo evento en el primer nodo.

El método 100 o 110 comprende una etapa 112 de transferencia, desde el primer nodo al segundo nodo de información indicativa de los retardos de eventos individuales (o un máximo de ellos) correspondiente a la pluralidad de eventos de entrada suscritos en el primer nodo de red. El método 100 o 110 comprende, además, una etapa 114 de transferencia de un contenido y/o estado del detector de eventos desde el primer nodo al segundo nodo con el fin de alcanzar a un detector de eventos objeto de migración en el segundo nodo. Según se explicó con anterioridad, el detector de eventos puede comprender una máquina de estados finitos que comprende un número finito de estados y algunas transiciones entre dichos estados. Es decir, el contenido del detector de eventos objeto de migración puede reflejar un estado actual de la máquina de estados del procesador como, a modo de ejemplo, variables individuales o conjuntos de variables.

Además, el método 100 o 110 comprende una etapa 116 de reenvío de la pluralidad de eventos de entrada desde el primer nodo al segundo nodo. Como será evidente para los expertos en esta técnica, las etapas 112, 114 y 116 pueden realizarse o ejecutarse por el propio primer nodo. En consecuencia, las etapas 112 a 116 forman juntas el método 110 que puede realizarse por el primer nodo de red o un dispositivo controlador central, que puede controlar la interacción y comunicación informática entre varios nodos de red.

El método de migración 100 o el sub-método 120 comprende, además, una etapa 122 de recepción, en el segundo nodo de red, de la pluralidad de eventos de entrada reenviados desde el primer nodo y, en paralelo, desde detectores de eventos origen asociados con la pluralidad de eventos de entrada. Es decir, en la etapa 122 cada uno de la pluralidad de eventos de entrada se recibe en el segundo nodo por intermedio de dos rutas. Una primera ruta es una ruta de desvío que lleva desde un detector de eventos origen de un evento de entrada respectivo por intermedio del primer nodo de red al segundo nodo de red. La segunda ruta es una ruta directa que lleva desde un detector de eventos origen de eventos de entrada "directamente" al segundo nodo. En consecuencia, el término "directamente" significa por intermedio de la red informática distribuida sin realizar el desvío a través del primer nodo. La etapa 122 es seguida por una etapa 124 de procesamiento ordenador, en el segundo nodo, de la pluralidad de eventos de entrada recibidos sobre la base de la información transferida sobre los retardos de eventos individuales (o su máximo) procedentes del primer nodo y sobre la base de los nuevos retardos de eventos individuales de entre la pluralidad de eventos de entrada directamente recibidos desde los detectores de eventos (origen) asociados. Es decir, la etapa 124 comprende también la derivación de un orden correcto de los eventos de entrada combinando la información de retardos desde los eventos que llegan a lo largo de la ruta de desvío o de nueva trayectoria operativa con información del retardo procedente de eventos que llegan en el segundo nodo a lo largo de la ruta directa.

Como será evidente para los expertos en esta técnica, las etapas 122 y 124 juntas constituyen el método 120 que puede realizarse en el segundo nodo de red del sistema informático distribuido. Con el fin de poder realizar las respectivas etapas de los métodos 100, 110 y 120, el sistema informático distribuido y/o el primer nodo de red y/o el segundo nodo de red pueden comprender respectivos medios para ejecutar las etapas individuales, respectivamente. En consecuencia, dichos medios como, a modo de ejemplo, un medio para transferir un medio para reenviar, un medio para recibir o un medio para procesar, pueden comprender circuitos analógicos o digitales eléctricos, que pueden programarse en consecuencia. A modo de ejemplo, un sistema informático distribuido puede comprender una pluralidad de unidades CPUs que funcionan como nodos de red individuales. En correspondencia, el primero y/o el segundo nodo de red pueden ser individuales, unidades CPUs separadas del sistema informático distribuido. En conformidad con algunas formas de realización, el sistema informático distribuido puede estar acoplado a una red de sensores (inalámbrica), en particular una red de procesamiento o localización, que se ha descrito en la parte introductoria. En caso de una red de sensores de localización, está última puede entregar datos de sensores o eventos sin procesamiento, que comprenden datos de posiciones así como datos cinemáticos, para el sistema informático distribuido. Los datos de sensores, sin procesamiento, pueden considerarse como el nivel más bajo de datos de eventos en el sistema informático distribuido. Pueden servir también para entregar la señal de reloj clk. Sobre la base de los datos de sensores de más bajo nivel, los así denominados eventos primitivos pueden detectarse como eventos del primer nivel. Es decir, los eventos primitivos solamente están basados en datos de sensores (localización) sin procesamiento. Todos los eventos de más alto nivel son derivados, luego, a partir de los eventos primitivos, lo que tiene la ventaja de que las tasas de datos extremadamente altas de los datos de sensores, sin procesamiento, pueden evitarse dentro del sistema informático distribuido. Lo que antecede puede permitir la

detección y procesamiento de eventos en tiempo real, tal como se requiere por la detección de eventos para acontecimientos deportivos de alta popularidad, tal como de fútbol, a modo de ejemplo. Es decir, en una forma de realización particular, el sistema informático distribuido puede utilizarse para detectar eventos, en un juego deportivo, tal como fútbol, rugby o fútbol americano.

5 Formas de realización de la presente invención se describirán a continuación, con más detalle, haciendo referencia a las Figuras 2 a 5.

10 Se considera una topología de red de un sistema informático distribuido 200 ilustrado en la Figura 2a. El sistema informático distribuido 200 a modo de ejemplo, comprende enrutadores 210 y 220. Cada expedidor 210, 200 está asociado a una pluralidad de nodos de red del sistema informático distribuido 200. A modo de ejemplo, el expedidor 210 está asociado a un primer nodo de red N_1 y a un segundo nodo de red N_2 . El segundo expedidor 220 está asociado con un tercer nodo de red N_3 así como a un cuarto nodo de red N_4 del sistema informático distribuido 200. Los enrutadores 210 y 220 están interconectados por intermedio de un enlace 230 (p.ej., un enlace de Ethernet) de modo que todos los nodos de red N_1 a N_4 estén interconectados entre sí.

15 La Figura 2a ilustra, a modo de ejemplo, una situación de red antes de que se realice una migración desde el nodo de red N_1 al nodo N_2 . Antes de su migración, un detector de eventos, a modo de ejemplo, funciona en el nodo N_1 y suscribe tres eventos de entrada A (editados o emitidos por el nodo N_3), B (editados por el nodo N_2) y C (editados por el nodo N_4). El detector de eventos objeto de migración desde el nodo N_1 al nodo N_2 genera, sobre la base de los eventos de entrada suscritos A, B y C, un evento compuesto D, que de nuevo se suscribe por el nodo de red N_3 . Dicho de otro modo, se proporciona un evento de entrada A al detector de eventos que funciona en la primera red N_1 , mediante el detector de eventos origen que funciona en el nodo N_3 (véase referencia numérica 242). El segundo evento de entrada B se proporciona al detector de eventos del nodo N_1 por el segundo nodo N_2 (véase referencia numérica 244). El tercer evento de entrada C se proporciona al detector de eventos que funciona en el nodo de red N_1 por el nodo de red N_4 (véase referencia numérica 246). El detector de eventos que funciona en el primer nodo N_1 es un detector de eventos origen para el evento compuesto D que se proporciona inicialmente desde el nodo de red N_1 a un detector de eventos de nivel superior que funciona en el nodo de red N_3 (véase referencia numérica 248).

20 Volviendo ahora a la Figura 2b en la que se ilustra un escenario operativo después de que el detector de eventos para el evento compuesto D haya sido objeto de migración o transferencia desde el primer nodo N_1 al segundo nodo N_2 haciendo uso del método 100. Cuando este detector de eventos es objeto de migración desde N_1 a N_2 , los nodos origen de los eventos de entrada suscritos A, B, C siguen siendo los mismos (p.ej., N_3 , N_2 , N_4) pero sus retardos de eventos asociados cambian de forma más definida. A modo de ejemplo, un retardo $\delta(B)$ de un evento de entrada B en el nuevo nodo N_2 puede contraerse debido a la migración, porque el evento de entrada B es ahora un evento local en el segundo nodo N_2 después de la migración. Los retardos de los otros eventos de entrada suscritos A y C pueden contraerse o no hacerlo. Conviene señalar que el retardo del evento compuesto D en el nodo de red N_3 puede cambiar también aun cuando el abonado en el nodo N_3 no participe en la migración.

25 A no ser que los retardos de eventos $\delta(.)$ se contraigan para todos los eventos implicados, una migración sin procesamiento es probable que falle porque el detector de eventos objeto de migración en el nodo N_2 o cualesquiera detectores de eventos en una capa superior ya no detecten los eventos suscritos A, B y C en un orden (temporal) correcto porque el valor de K que funcionaba bien en el nodo antiguo (N_1) puede ser demasiado pequeño para el nuevo nodo (N_2) y los nuevos retardos de eventos $\delta'(.)$ después de la migración.

30 Formas de realización de la presente invención tienen como objetivo garantizar eventos de entrada ordenados para el detector de eventos objeto de migración midiendo y adaptando los retardos de eventos. En condiciones normales, puede ser que no sea posible hacer funcionar los detectores de eventos antiguo y nuevo en paralelo si la sobrecarga de la unidad CPU o el desbordamiento de la capacidad de la memoria intermedia han iniciado operativamente la migración. Además, el reenvío del flujo de datos completo desde el nodo antiguo al nodo nuevo es prohibitivo puesto que puede causar altas cargas de red y sobrecargas de procesamiento durante un periodo de tiempo prolongado, si se producen eventos particulares con poca frecuencia.

35 Formas de realización de la presente invención pueden efectuar la migración o transferencia de detectores de eventos en el tiempo de ejecución e inicializar los valores de K en conformidad con los retardos de eventos antiguos en el nuevo nodo. La latencia introducida es despreciable y ambas instancias de detectores de eventos antiguos y nuevos así como detectores de eventos de nivel superior pueden detectar eventos en orden en cualquier momento. El detector de eventos antiguo puede interrumpir su funcionamiento tan pronto como su información de estado o contenido se haya copiado/transferido al nuevo nodo de red y las sobrecargas de gestión de redes pueden mantenerse en una magnitud mínima.

40 Una idea inventiva subyacente a las formas de realización de la presente invención es que es para una migración cooperativa o transferencia del detector de eventos desde un nodo a otro, el nuevo nodo de red no solamente suscribe los eventos de entrada necesarios desde sus detectores de eventos origen, sino que el nodo antiguo reenvía también dichos eventos de entrada al nuevo nodo. De esta manera, el nuevo nodo de cálculo informático puede derivar el orden temporal correcto (o valor K) combinando la información de retardos de eventos precedente

de los eventos de entrada que llegan a lo largo de dos rutas diferentes (ruta de desvío por intermedio del nodo antiguo y ruta directa desde las fuentes de eventos de entrada).

5 La Figura 3 ilustra un diagrama de secuencia de mensajes (MSC) 300 de un proceso de migración de detectores de eventos, a modo de ejemplo, en conformidad con una forma de realización.

10 Como puede deducirse del diagrama de secuencias de mensajes 300, el nodo de red N_2 puede suscribir en primer lugar (véase referencia numérica 302) a los eventos de entrada A, B y C del proceso u objeto del detector de eventos todavía a ejecutarse en el primer (antiguo) nodo (N_1). En este caso, la suscripción 302 a los eventos de entrada A, B y C tiene lugar en un instante temporal t_0 . La suscripción 302 puede, a modo de ejemplo, tener lugar en respuesta a una instrucción de migración, que ha sido recibida desde un controlador de red central en el nodo de red N_2 . Dicha entidad de controlador central puede tener información sobre más o la totalidad de los retardos de señales de eventos que se desplazan a lo largo de diferentes rutas en el sistema distribuido. De este modo, un retado de un evento puede a modo de ejemplo, deberse a diferentes fluctuaciones, capacidades de funcionamiento y/o tiempos de propagación de señales de las diferentes rutas o encaminamiento de redes diferentes. Como alternativa o de forma adicional, el controlador central puede tener información sobre más o la totalidad de las situaciones de carga (p.ej., carga de la unidad CPU) de nodos individuales y/o información indicativa de una carga de sistema global. En consecuencia, en conformidad con algunas formas de realización, un aparato para la migración puede tener conocimiento de parámetros del sistema globales. En conformidad con otras formas de realización, el aparato para efectuar la migración puede ser también no central, esto es, estar distribuido a través de una pluralidad de nodos de red. En este caso, un nodo de red puede tener solamente información limitada indicativa de su propia carga o situación de retado de eventos. Por lo tanto, puede no estar disponible, en algunas formas de realización, el conocimiento de parámetros de sistemas globales.

25 El controlador MSC 300 supone también que se ha creado ya una instancia del detector de eventos (instanciados) en el nuevo nodo de red N_2 . Es decir, un aparato para efectuar la migración puede comprender también medios para crear una instancia de proceso u objeto, que esté en correspondencia con el primer detector de eventos, en el segundo nodo con el fin de obtener el segundo detector de eventos. Ha de entenderse una instancia operativa como una ocurrencia o una copia de un objeto, con ejecución en curso o no. Las instancias de una clase o de un objeto comparten el mismo conjunto de atributos, que normalmente diferirán en qué contienen dichos atributos.

35 Además, en el instante temporal t_0 o inmediatamente después de dicho tiempo t_0 , el segundo nodo de red N_2 puede enviar un mensaje de demanda de transferencia al primer nodo N_1 (véase referencia numérica 304). En respuesta al último mensaje de demanda de transferencia, el nodo de red antiguo N_1 puede iniciar el reenvío de la pluralidad de eventos de entrada A, B y C desde el primer nodo N_1 al segundo nodo N_2 (véase referencia numérica 316). Además, el primer nodo N_1 puede dar respuesta al mensaje de demanda de transferencia de la etapa 304 con dos mensajes. Un primer mensaje transferido desde el nodo antiguo N_1 al nodo nuevo N_2 puede contener la información de retado actual para cada evento de entrada suscrito A, B y C (véase etapa 312). Es decir, información sobre los retardos de eventos individuales correspondientes a la pluralidad de eventos de entrada A, B y C se transfieren desde el primer nodo N_1 al segundo nodo N_2 en la etapa 312. A modo de ejemplo, el nodo N_1 puede recibir los eventos operativos de entrada con retardos de eventos $\delta(A) = 30$ ms, $\delta(B) = 10$ ms y $\delta(C) = 20$ ms, de modo que $K = \max(\delta(A), \delta(B), \delta(C)) = 30$ ms. De este modo, los retardos $\delta(\cdot)$ pueden incluir ya los márgenes de seguridad respectivos. Por lo tanto en conformidad con algunas formas de realización, los valores de retardos individuales $\delta(A)$, $\delta(B)$, $\delta(C)$ y/o el valor de K resultante, $K = \max(\delta(A), \delta(B), \delta(C))$ pueden transferirse desde el nodo N_1 al nodo N_2 por intermedio del mensaje 45 312. El paquete o mensaje de información de retardos 312 puede contener también la marca temporal actual t_s de modo que el segundo nodo N_2 pueda determinar $d_f = \text{clk} - t_s$, es decir, el sub-retado d_f de reenvío de un evento desde el nodo N_1 al nodo N_2 . Se supone que d_f tiene el valor de 5 ms para el ejemplo actual. Dicho de otro modo, la transferencia de la información de retado de eventos de los eventos de entrada A, B y C desde el primer nodo N_1 al segundo nodo N_2 puede comprender la transferencia de una marca temporal de transferencia t_s que es indicativa de un instante temporal en el que dicha información se ha enviado desde el primer nodo N_1 al segundo nodo N_2 . Además, la transferencia de la información de retado de eventos puede comprender la determinación, en el segundo nodo N_2 , de un retado de transferencia común d_f desde el primer nodo N_1 al segundo nodo N_2 , en donde el retado de transferencia común d_f es indicativo de una diferencia temporal entre la marca temporal de transferencia y una recepción o tiempo de llegada de la información sobre los retardos de eventos en el segundo nodo N_2 . Disponiendo de la información del retado del mensaje 312, el segundo nodo N_2 puede procesar la pluralidad de eventos de entrada recibidos A, B y C, inicialmente sobre la base de a información de retardos de eventos de entrada transferidos ($\delta(A)$, $\delta(B)$, $\delta(C)$ y/o K) y el retado de transferencia común.

60 El segundo mensaje o paquete antes citado, que puede transferirse desde el nodo antiguo N_1 al nodo nuevo N_2 en la etapa 314 comprende una toma instantánea (esto es, una copia del estado actual) del detector de eventos. Dicho de otro modo, en la etapa 314, un contenido del detector de eventos puede transferirse desde el primer nodo N_1 a la instancia del detector de eventos del segundo nodo N_2 para obtener un detector de eventos objeto de migración en el segundo nodo N_2 .

65 A modo de ejemplo, los servicios de sistemas operativos pueden comprender funcionalidades de copia o de

transferencia, a las que pueden tener acceso un medio de transferencia para transferir el contenido. Es decir, un medio de transferencia puede comprender medios para acceder a las funcionalidades de copia de proceso u objeto de un sistema operativo que se ejecuta en los nodos de cálculo informático o en alguna otra entidad controladora del sistema distribuido. Los medios de transferencia pueden ser también utilizables para “congelar” el contenido del primer detector de eventos en un instante temporal predefinido con el fin de obtener una condición definida del primer detector de eventos en el instante temporal de la ‘congelación’. Dicho de otro modo, la transferencia del contenido del detector de eventos desde el primer nodo N_1 al segundo nodo N_2 puede comprender la denominada congelación de un estado actual del detector de eventos del primer nodo N_1 y la copia del estado ‘congelado’ a una instancia del detector de eventos del segundo nodo N_2 . Por lo tanto, el término “congelación” significa que el detector de eventos se lleva a un estado determinado y deja de dar respuesta a sus entradas en el instante de la congelación. Dicho de otro modo, el primer detector de eventos puede interrumpir su funcionamiento en el instante de congelación con el fin de copiar su contenido o condición de estado para el segundo detector de eventos instanciados.

Por lo tanto, la transferencia de contenidos 314 puede comprender la copia del contenido/estado en condición de congelación desde el primer detector de eventos al segundo detector de eventos con el fin de obtener el mismo estado determinado o condición definida en el segundo detector de eventos del nodo N_2 .

Antes del segundo recientemente instanciado, esto es, el detector de eventos se inicia o activa en el segundo nodo N_2 , pudiéndose los eventos de entrada (idénticos) de ambos detectores de eventos ser objeto de memorización intermedia de modo que ambos detectores de eventos que, debido a la operación de copia, tienen ya idealmente la misma condición interna definida, obtengan también datos de entrada idénticos que se inician desde el instante temporal predefinido, esto es, el instante de la “congelación”. Puesto que los relojes pueden variar entre ambos nodos, el nodo N_1 debe cerciorarse de que el nodo N_2 puede memorizar todos los eventos de modo que pueda recuperar el estado correcto original desde la toma instantánea transferida. A modo de ejemplo, si el nodo N_1 no ha procesado un evento de entrada desde el instante temporal t_0 , su estado actual puede utilizarse para la toma instantánea y se tiene $t_{\text{snap}} = t_0$. De no ser así, t_{snap} puede establecerse para la marca temporal de ocurrencia del último evento de entrada procesado del nodo N_1 . Después de enviar la toma instantánea en la etapa 314, el primer nodo de red N_1 puede terminar operativamente el detector de eventos puesto que el nuevo nodo N_2 puede asumir e iniciar ahora la obtención de la copia del detector de eventos original, lo que se indica por la referencia numérica 315. Después del cierre del detector de eventos en el nodo antiguo N_1 y la iniciación de la instancia del detector de eventos en el nuevo nodo N_2 , el nodo antiguo N_1 puede continuar reenviando los eventos de entrada A, B y C (véase la referencia numérica 316) al nuevo nodo N_2 hasta que este último haya acabado un procedimiento de adaptación de retardos de eventos 320 que se describirá a continuación.

A modo de ejemplo, en el instante temporal $t_0 = \text{clk} = 100$, el segundo (esto es, nuevo) nodo N_2 de red envía el mensaje de demanda de transferencia 304 al nodo de red antiguo N_1 . Puesto que el nodo de red antiguo N_1 está procesando continuamente eventos de entrada suscritos A, B y C, puede estar ya ocupado con un evento de entrada que tiene la marca temporal 180. Cuando el mensaje de demanda de transferencia llega procedente del nodo N_2 , el nodo de red antiguo N_1 puede realizar la toma instantánea del detector de eventos original, establecer t_{snap} a 180 (correspondiente a la marca temporal del evento de entrada procesado en último lugar), enviar el mensaje de toma instantánea 314 a un nuevo nodo de red N_2 y terminar la acción del detector de eventos original en el nodo de red N_1 . El nuevo nodo de red N_2 puede establecer el estado de la nueva instancia del detector de eventos y procesar cualesquiera eventos de entrada memorizados A, B y C con una marca temporal sobre t_{snap} (esto es, 180) solamente. Es decir, el reenvío de los eventos de entrada A, B y C desde el nodo de red antiguo N_1 al nuevo nodo de red N_2 comprende la memorización intermedia, en el nuevo nodo N_2 , de todos los eventos de entrada reenviados y/o recibidos A, B y C, comenzando en el tiempo t_0 del mensaje de demanda de transferencia 304 hasta el tiempo de la toma de instantánea t_{snap} . Cuando el detector de eventos objeto de migración en el nuevo nodo N_2 comienza a funcionar, puede procesar un evento memorizado que tenga una marca temporal, esto es, un tiempo de ocurrencia, superior al tiempo de toma instantánea t_{snap} . Dicho de otro modo, la recepción de la pluralidad de eventos de entrada A, B y C reenviados desde el primer nodo N_1 puede comprender, en el nuevo nodo N_2 , la memorización instantánea de todos los eventos de entrada reenviados A, B y C hasta el tiempo de toma instantánea t_{snap} cuando el contenido/estado ‘congelado’ del detector de eventos sea transferido desde el primer nodo N_1 al segundo nodo N_2 y en donde el procesamiento de la pluralidad de los eventos operativos A, B, C recibidos en el segundo nodo N_2 comprende el procesamiento de los eventos memorizados que tengan una marca temporal de evento o tiempo de ocurrencia superior al tiempo de toma instantánea t_{snap} .

A continuación, se describirá, en más detalle, la adaptación de la información del retado de evento ($\delta(A)$, $\delta(B)$, $\delta(C)$ y K) de la etapa 320. Durante la etapa 320, los retardos de eventos transferidos $\delta(A)$, $\delta(B)$, $\delta(C)$ desde el nodo antiguo N_1 y/o el valor de K resultante pueden posteriormente adaptarse o actualizarse. Además, se cancelan los ecos operativos de eventos de entrada suscritos que se reciben en el segundo nodo N_2 por intermedio de la ruta de desvío desde el nodo antiguo N_1 . Con las formas de realización, es posible desplazar correctamente un detector de eventos en funcionamiento desde un primer nodo de red N_1 a un segundo nodo de red N_2 . No obstante, en el nuevo nodo de red N_2 , los retardos de eventos de los eventos de entrada suscritos A, B y C y por lo tanto, los valores de K adecuados pueden ser diferentes, puesto que los retardos de los eventos suscritos pueden haber cambiado debido a la migración y la recepción directa resultante. En lugar de calcular la magnitud de la memoria intermedia K_D para

el evento compuesto D desde el principio en el segundo nodo N_2 , dicha capacidad de memoria K'_D puede inicializarse en conformidad con la información de retado de evento recibida desde un nodo antiguo N_1 por intermedio del mensaje de información de retardos 312. Es decir, el retado máximo o la magnitud de memoria K'_D en el nuevo nodo, puede determinarse en conformidad con

5

$$K'_D = K'_D + d_f$$

$$= \max(\delta(A) + d_f, \delta(B) + d_f, \delta(C) + d_f)$$

10

$$= \max(30+5, 10+5, 20+5) = 35 \text{ ms,}$$

en donde K_D indica la capacidad de la memoria intermedia del evento compuesto D en el nodo antiguo N_1 .

15

Es decir, la etapa 124 de procesamiento de la pluralidad de eventos de entrada recibidos A, B y C en el segundo nodo N_2 puede comprender la determinación de un tiempo de salida t_{out} para retransmitir un evento de entrada recibido hacia el detector de eventos objeto de migración sobre la base de su tiempo de detección u ocurrencia de evento respectivo y un retardo de eventos máximo K'_D de la pluralidad de eventos de entrada A, B y C.

20

Según se ilustra en la Figura 4, el nuevo nodo de red N_2 recibe todos los eventos de entrada (A, B y C) dos veces. En general, un evento de entrada e alcanza el nuevo nodo N_2 en una ruta directa 402 con un retado $\delta(e)$ y alcanza también el nuevo nodo N_2 en una ruta de desvío 404 por intermedio del nodo de red antiguo N_1 . De este modo, la ruta directa 402 alcanza desde un nodo origen o fuente N_3 por intermedio de la red distribuida 406 directamente hacia el nodo de destino N_2 , mientras que la ruta de desvío alcanza desde el nodo origen N_3 por intermedio de la red 406 al nodo antiguo N_1 y desde este último al nuevo nodo N_2 . Tan pronto como el nuevo nodo N_2 recibe un evento de entrada e, a lo largo de la ruta directa 402, puede actualizar su capacidad de memorización K utilizando $\delta'(e)$ en lugar de $\delta(e) + d_f$. Es decir, la etapa de procesamiento 124 comprende también la actualización de la información sobre los retardos de eventos individuales sobre la base de los retardos de eventos individuales de la pluralidad de eventos de entrada recibidos desde los detectores de eventos asociados por intermedio de la ruta directa 402. Es decir, la información sobre un retado de evento individual correspondiente a un evento de entrada en el primer nodo puede actualizarse tan pronto como dicho evento de entrada sea recibido en el segundo nodo por intermedio de una ruta directa 402 desde el receptor origen N_3 asociado al evento de entrada e, de modo que el correspondiente retado de evento antiguo $\delta(e) - d_f$ se sustituye por una diferencia entre la ocurrencia del evento o un tiempo de detección en el detector de eventos origen N_3 y una recepción de eventos o tiempo de llega del evento respectivo e en el segundo nodo N_2 .

35

A modo de ejemplo, un nuevo retado (directo) de $\delta(A) = 25$ ms para el primer evento A directamente recibido reduciría la capacidad de memorización requerida K_D desde 35 ms a máximo $(25, 15, 25) = 25$ ms. Un valor de K actualizado puede anunciarse desde el segundo nodo N_2 para la suscripción de detectores de eventos en niveles de jerarquía de eventos más alto o superior. Es decir, en caso de que la actualización de la información del retado de evento transferida lleve a un retardo de eventos máximo actualizado o un valor de memorización (valor K) de la pluralidad de eventos de entrada (p.ej., A, B, C), dicho retardo de eventos máximo actualizado o valor de memorización puede comunicarse a detectores de eventos de capa más alta o de flujo descendente, que estén utilizando un evento de salida D del detector de eventos objeto de migración del segundo nodo N_2 como un evento de entrada. Dicho de otro modo, esto significa que el valor K actualizado puede comunicarse a detectores de eventos de capa superior que se han suscrito al evento compuesto D del detector de eventos objeto de migración.

40

45

50

La recepción de los eventos de entrada en el nuevo nodo dos veces puede ser ventajoso operativamente para inicializar la nueva capacidad K de memorización intermedia, con los eventos de entrada objeto de eco operativo pudiendo plantear problemas para un detector de eventos objeto de migración. Para cerciorarse de que una unidad de ordenamiento de eventos asociada con el detector de eventos objeto de migración solamente detecte un evento de entrada a la vez, necesita eliminarse uno de los eventos objeto de eco operativo. Para cada evento de tipo A, B, C, habrá una primera vez en que el nuevo nodo N_2 detecte un evento por intermedio de la ruta directa 402 y su eco operativo a través de la ruta de desvío 404. Antes de ese momento, el evento de entrada con el retado inferior se transmite a lo largo de la unidad de ordenamiento, mientras que la más reciente versión recibida del evento de entrada es eliminada. Tan pronto como sea recibida la versión de evento directa, es decir, la versión de e' del evento recibida por intermedio de la ruta directa 402, el nodo antiguo N_1 puede notificarse de que este tipo de evento particular no ha de reenviarse ya desde el nodo antiguo N_1 al nodo nuevo N_2 . Es decir, el segundo nodo N_2 puede ser utilizable para informar al primer nodo N_1 de la interrupción del envío de un evento de entrada e tan pronto como el segundo nodo N_2 haya recibido el evento de entrada respectivo e por intermedio de la ruta directa 402 desde un detector de eventos origen (p.ej., N_3) del sistema informático distribuido 406. Un emisor de un evento puede, a modo de ejemplo, determinarse mediante la información del emisor incluida por los paquetes recibidos.

55

60

Volviendo ahora a la Figura 5, se ilustra un pseudocódigo (abreviado) 500 de la adaptación del retado y cancelación de ecos operativos descritos, en conformidad con una forma de realización de la presente invención.

65

El pseudocódigo ilustrado 500 comienza con una inicialización de un eco de objeto que tipo *EventList* (véase referencia numérica 502). A continuación, en una programación denominada *for-loop* 504, los retardos de eventos d del detector de eventos objeto de migración son inicializados con los retardos de eventos transferidos desde el primer nodo más el valor de retardo de transferencia común d_f . En una etapa 506, el valor K del detector de eventos nuevo u objeto de migración ED_{new} se determina en función de los retardos máximos $\max(\text{delays})$. La primera declaración de programación *if* 510 dentro del bucle de *while* 508 comprueba si un indicador ID de un evento recibido pertenece, o no, a la cantidad de eventos suscritos. Si no es así, en tal caso, la recepción de eventos se continúa hasta que se reciba un evento de entrada suscrito e . Si el evento de entrada suscrito e ha sido recibido mediante la ruta directa 402 (véase bucle de *if* 512), entonces el retardo de eventos del evento e que tiene el *id* está actualizado al valor clk-e.ts (véase referencia numérica 514). Posteriormente, el valor de K se actualiza en la línea 516. En caso de que el valor actualizado de K valor K_n no sea igual al valor de K anteriormente determinado, el valor de K del detector de eventos nuevo u objeto de migración ED_{new} se actualiza al nuevo valor K_n de K . Además, el valor de K cambiado se notifica a los detectores de eventos de la capa más alta (véase declaración de *if* 518). Además, el nodo antiguo N_1 es notificado de que no es necesario continuar el reenvío del evento e que tiene el identificador *id* (véase línea 520). En la siguiente declara de *if* 522, se suprime o borra un eco del evento e en caso de que esté presente. De no ser así, se añade el eco operativo (véase referencia numérica 524) el evento recibido e y se lleva a la unidad de ordenamiento en asociación con la instancia de detector de eventos nueva u objeto de migración ED_{new} .

El concepto de migración o transferencia cooperativa en conformidad con las formas de realización de la presente invención puede realizar una migración segura de detectores de eventos desde un primer nodo a un segundo nodo de un sistema informático distribuido y pueden asegurar simultáneamente un orden de eventos total para el nuevo detector de eventos. Los receptores pueden copiarse e inmediatamente desactivarse porque el detector de eventos objeto de migración calibra, de forma interactiva, sus valores de K . Las formas de realización de la presente invención pueden conseguir un procesamiento de eventos distribuido fiable de flujos de sensores de altas tasas de datos incluso bajo la predominancia de eventos fuera de orden. Los retardos de eventos pueden medirse y adaptarse en el tiempo de ejecución y los eventos pueden posponerse en tanto que sea necesario para garantizar un orden total en los flujos de eventos entrantes. No se requiere ningún conocimiento *a priori* de los retardos de eventos. Para equilibrar la carga y para utilizar mejor una potencia de cálculo informático disponible, los detectores de eventos pueden efectuar la migración en el tiempo de ejecución mientras que las limitaciones de temporización se sigan manteniendo válidas.

Las formas de realización de la presente invención son adecuadas para un sistema de localización en tiempo real (RTLS) en una aplicación deportiva de fútbol. Dicho sistema RTLS puede, a modo de ejemplo, efectuar el seguimiento de 144 transmisores al mismo tiempo en 2000 puntos de muestreo por segundo para un balón y 200 puntos de muestreo por segundo para jugadores y árbitros. Cada jugador puede estar provisto de cuatro transmisores, uno en cada uno de sus miembros. Los datos de sensores, sin procesamiento, procedentes de los transmisores comprenden posiciones absolutas en milímetros, velocidad, aceleración y calidad de localización (QoL) para cualquier dirección. El acontecimiento deportivo de fútbol necesita estas tasas de muestreo. Con 2000 puntos de muestreo por segundo para el balón una velocidad de hasta 150 km/h, dos posiciones sucesivas pueden estar separadas en una distancia de 2 cm. Puesto que los eventos del fútbol tales como pase, doble pase, disparo a gol, etc., suceden dentro de una fracción de segundo, el procesamiento y migración de eventos debe asegurar que los eventos sean detectados a su debido tiempo de modo que una jerarquía de detectores de eventos pueda ayudar a un observador humano, a modo de ejemplo, un informador deportivo, para trabajar con la salida en directo del sistema.

La descripción y los dibujos ilustran simplemente los principios de la invención. Por ello, serán apreciados por los expertos en esta técnica que serán capaces de realizar varias disposiciones que, aunque no se describen aquí explícitamente ni se ilustran, materializan los principios de la invención y están incluidos dentro de su alcance. Además, todas las realizaciones, a modo de ejemplo, aquí descritas están principalmente previstas expresamente para ser solamente para fines pedagógicos para ayudar al lector en el entendimiento de los principios de la invención y los conceptos aportados por los inventores para mejorar esta técnica y han de interpretarse como no limitándose a los ejemplos y condiciones concretamente descritos. Además, todas las declaraciones aquí dadas con respecto a los principios, aspectos y formas de realización de la invención, así como sus realizaciones específicas a modo de ejemplo, se pretenden para abarcar sus equivalentes.

Los bloques funcionales indicados como “medios para...” (que realizan una determinada función) deberán entenderse como bloques funcionales que comprenden circuitos que están adaptados, configurados o utilizables para realizar una determinada función, respectivamente. En consecuencia, un término como “medios para algo” pueden entenderse también como un “medio que está adaptado, configurado o que es utilizable para hacer algo”. Un medio que esté adaptado para realizar una determinada función, por lo tanto, no implica que dicho medio esté realizando necesariamente dicha función (en un instante temporal dado).

Las funciones de varios elementos que se ilustran en las Figuras, incluyendo cualesquiera bloques funcionales pueden proporcionarse mediante el uso de hardware dedicado, p.ej., un procesador, así como un hardware capaz de ejecutar un programa informático en asociación con un software adecuado. Cuando se proporcionan por un

5 procesador, las funciones pueden proporcionarse por un procesador dedicado único, por un procesador compartido único o por una pluralidad de procesadores individuales, algunos de los cuales pueden ser compartidos. Además, el uso explícito del término “procesador” o “controlador” no debe interpretarse para referirse exclusivamente a hardware capaz de ejecutar un programa informático y pueden incluir implícitamente, sin limitación, hardware de procesador de señal digital (DSP), procesador de red, circuito integrado específico de la aplicación (ASIC), conjunto de puertas programables in situ (FPGA), memoria de lectura solamente (ROM) para software de memorización, memoria de acceso aleatorio (RAM) y memorización no volátil. Otros equipos físicos, convencionales y/o personalizados, pueden incluirse también a este respecto.

10 Debe apreciarse por los expertos en esta técnica que cualesquiera diagramas de bloques de esta descripción representan vistas conceptuales de circuitos ilustrativos que materializan los principios de la invención. De forma similar, se apreciará que cualesquiera gráficos de flujo, diagramas de flujo, diagramas de transición de estados, pseudocódigo y elementos similares representan varios procesos que pueden representarse esencialmente en un soporte legible por ordenador y ejecutarse por un ordenador o procesador, si dicho ordenador o procesador está explícitamente ilustrado o no lo está.

15 Además, las siguientes reivindicaciones se incorporan aquí en la descripción detallada, en donde cada reivindicación puede constituir por sí misma una forma de realización separada. Aunque cada reivindicación puede considerarse por sí misma como una forma de realización separada, conviene señalar que aunque una reivindicación subordinada pueda referirse en las reivindicaciones a una combinación específica con una o más otras reivindicaciones, otras formas de realización pueden incluir también una combinación de la reivindicación subordinada con el contenido de cualquier otra reivindicación subordinada. Dichas combinaciones son aquí propuestas a no ser que no esté prevista que se declare una combinación específica. Además, está previsto incluir también características establecidas en una reivindicación para cualquier otra reivindicación independiente aun cuando esta reivindicación no se realice directamente con dependencia de la reivindicación independiente.

20 Además, conviene señalar que los métodos dados a conocer con anterioridad en la memoria o en las reivindicaciones pueden ponerse en práctica mediante un dispositivo que tenga medios para realizar cada una de las respectivas etapas de estos métodos.

30 Además, ha de entenderse que la revelación de múltiples etapas o funciones dadas a conocer en la memoria o reivindicaciones no pueden interpretarse como que están dentro del orden específico. Por lo tanto, la revelación de múltiples etapas o funciones no limitará a éstas a un orden particular a no ser que dichas etapas o funciones no sean intercambiables por razones técnicas.

35 Además, en algunas formas de realización una etapa única puede incluir o puede descomponerse en múltiples sub-etapas. Dichas sub-etapas pueden incluirse y formar parte de la idea inventiva de esta etapa única a no ser que se excluya de forma explícita.

40

REIVINDICACIONES

1. Un método (100; 300) que permite la migración de un detector de eventos operativos de un primer nodo (N_1) de un sistema informático distribuido (200) hacia un segundo nodo (N_2) del sistema informático distribuido, en donde el detector de eventos operativos procesa una pluralidad de eventos operativos de entrada (e; A, B, C) para generar al menos un evento operativo de salida (D) sobre la base de los eventos operativos de entrada en donde cada uno de los eventos operativos de entrada está asociado a un retardo de evento operativo individual correspondiente a un tiempo de propagación del evento operativo de entrada hacia el primer nodo (N_1), cuyo método (100; 300) comprende:
- la transferencia (112; 312), desde el primer nodo (N_1) hacia el segundo nodo (N_2), de información relativa a los retardos de eventos operativos que corresponden a la pluralidad de eventos operativos de entrada (e; A, B, C) en el primer nodo (N_1);
- la transferencia (114; 314) del contenido del detector de eventos operativos desde el primer nodo (N_1) al segundo nodo (N_2) para obtener un detector de eventos operativos objeto de migración en el segundo nodo;
- la transmisión (116; 316) de la pluralidad de eventos operativos de entrada (e; A, B, C) del primer nodo (N_1) hacia el segundo nodo (N_2);
- la recepción (122), en el segundo nodo (N_2), de la pluralidad de eventos operativos de entrada (e; A, B, C) transmitida desde el primer nodo (N_1) y, en paralelo, a partir de detectores de eventos operativos de origen asociados a dicha pluralidad de eventos operativos de entrada; y
- el procesamiento (124; 320) en el segundo nodo (N_2), de la pluralidad de eventos operativos de entrada recibidos (e; A, B, C) sobre la base de la información transferida sobre los retardos de eventos operativos y sobre la base de los retardos de eventos operativos de la pluralidad de eventos operativos de entrada (e; A, B, C) recibidos desde los detectores de eventos operativos de origen asociados, en donde el procesamiento (124; 320) de la pluralidad de eventos operativos de entrada (e; A, B, C) recibidos en el segundo nodo (N_2) comprende la determinación de un tiempo para retransmitir un evento operativo de entrada recibido al detector de eventos operativos objeto de migración sobre la base de su tiempo de ocurrencia operativa y un retraso de evento operativo máximo (K'_D) de la pluralidad de eventos operativos de entrada, en donde el valor de retardo del evento operativo máximo (K'_D) está basado en la información transferida sobre los retardos de eventos operativos y sobre la base de los retardos de eventos operativos de la pluralidad de eventos operativos de entrada (e; A, B, C) recibidos desde los detectores de eventos operativos de origen asociados.
2. El método (100; 300) según la reivindicación 1, en donde la transferencia (112; 312) de la información sobre los retardos de eventos operativos de los eventos de entrada desde el primer nodo (N_1) hacia el segundo nodo (N_2) comprende la transferencia de una marca temporal de transferencia (t_s) que es indicativa de un tiempo en el que dicha información ha sido enviada desde el primer nodo al segundo nodo.
3. El método (100; 300) según la reivindicación 2, en donde la transferencia (112; 312) de la información sobre los retardos de eventos operativos comprende la determinación, en el segundo nodo (N_2), de un retardo de transferencia común (d_f) desde el primer nodo (N_1) al segundo nodo (N_2), en donde el retardo de transferencia común (d_f) es indicativo de una diferencia temporal entre la marca temporal de transferencia (t_s) y un tiempo de recepción de la información sobre los retardos de eventos operativos en el segundo nodo (N_2).
4. El método (100; 300) según la reivindicación 3, en donde el procesamiento (124; 320) de la pluralidad de eventos operativos de entrada recibidos (e; A, B, C) en el detector de eventos operativos objeto de migración está basada inicialmente en la información sobre los retardos de eventos operativos en el primer nodo (N_1) y el retardo de transferencia (d_f) común.
5. El método (100; 300) según una de las reivindicaciones precedentes, en donde la transferencia (114; 314) del contenido del detector de eventos operativos desde el primer nodo (N_1) al segundo nodo (N_2) comprende la congelación operativa de un estado actual del detector de eventos operativos del primer nodo (N_1) y la copia del estado de congelación operativa hacia una instancia del detector de eventos operativos del segundo nodo (N_2).
6. El método (100; 300) según una de las reivindicaciones precedentes, en donde la recepción (122) de la pluralidad de eventos operativos de entrada (e; A, B, C) reenviados desde el primer nodo (N_1) comprende, en el nuevo nodo (N_2) la memorización intermedia de todos los eventos operativos de entrada reenviados (e; A, B, C) hasta un momento de toma instantánea (t_{snap}) cuando el contenido del detector de eventos operativos se transfiere desde el primer nodo (N_1) al segundo nodo (N_2) y en donde el procesamiento (124; 320) de la pluralidad de eventos operativos de entrada recibidos (e; A, B, C) en el segundo nodo (N_2) comprende el procesamiento de eventos operativos memorizados que tienen un tiempo de aparición del evento operativo superior al tiempo de toma instantánea (t_{snap}).

7. El método (100; 300) según una de las reivindicaciones precedentes, en donde la información sobre un retardo de evento operativo correspondiente a un evento operativo de entrada (e; A, B, C) en el primer nodo (N_1) se actualiza tan pronto como dicho evento operativo de entrada es recibido en el segundo nodo (N_2) por intermedio de una ruta directa (402) desde el detector de eventos operativos de origen (N_3) asociado al evento operativo de entrada (e; A, B, C), de modo que el retardo de evento operativo correspondiente en relación con el primer nodo (N_1) sea sustituido por un retardo de evento operativo relacionado con la ruta directa (402) del evento operativo de entrada (e; A, B, C) hacia el segundo nodo (N_2).
8. El método (100; 300) según la reivindicación 7, en donde el procesamiento (124; 320) de la pluralidad de los eventos operativos de entrada (e; A, B, C) recibidos en el segundo nodo (N_2) comprende la actualización de la información transferida sobre un retardo de evento operativo individual y, en caso de que dicha actualización diera lugar a un valor de retardo de evento operativo máximo (K) actualizado de la pluralidad de eventos operativos de entrada (e; A, B, C), dicho valor de retardo de evento operativo máximo actualizado (K) se comunica a detectores de eventos que tienen una salida del detector de eventos operativos objeto de migración del segundo nodo (N_2) como una entrada.
9. El método (100; 300) según una de las reivindicaciones precedentes, en donde el segundo nodo (N_2) informa al primer nodo (N_1) para interrumpir el reenvío de un evento operativo de entrada tan pronto como el segundo nodo (N_2) haya recibido el respectivo evento operativo de entrada por intermedio de una ruta directa (402) desde un detector de eventos operativos de origen del sistema informático distribuido (200).
10. El método (100; 300) según una de las reivindicaciones precedentes, en donde un evento operativo es un evento primitivo que está directamente basado en datos de sensores o un evento compuesto, que está basado en eventos primitivos (e; A, B, C).
11. El método (100; 300) según una de las reivindicaciones precedentes, en donde los eventos operativos de entrada (e; A, B, C) son originalmente detectados en nodos distribuidos del sistema informático distribuido (200) y en donde los eventos operativos de entrada están basados en flujos de datos de sensores en paralelo de un sistema de localización geográfica.
12. Un método (120) para efectuar una migración de un detector de eventos operativos de un primer nodo (N_1) de un sistema informático distribuido (200) hacia un segundo nodo (N_2) del sistema informático distribuido, en donde el detector de eventos operativos procesa una pluralidad de eventos operativos de entrada (e; A, B, C) para generar al menos un evento operativo de salida (D) sobre la base de los eventos operativos de entrada en donde cada uno de los eventos de entrada está asociado con un retardo de evento individual que corresponde a un tiempo de propagación del evento operativo de entrada hacia el primer nodo (N_1), cuyo método (120) comprende:
la recepción (122), en el segundo nodo (N_2), de la pluralidad de eventos operativos de entrada (e; A, B, C) reenviados desde el primer nodo (N_1) y, en paralelo, desde detectores de eventos de origen asociados a la pluralidad de eventos operativos de entrada; y
el procesamiento (124; 320) en el segundo nodo (N_2) de la pluralidad de eventos operativos de entrada recibidos (e; A, B, C) sobre la base de la información transferida sobre los retardos de eventos y sobre la base de los retardos de eventos de la pluralidad de eventos de entrada (e; A, B, C) recibidos desde los detectores de eventos de origen asociados, en donde el procesamiento (124; 320) de la pluralidad de eventos de entrada recibidos (e; A, B, C) en el segundo nodo (N_2) comprende la determinación de un tiempo para retransmitir un evento de entrada recibido al detector de eventos objeto de migración sobre la base de su tiempo de aparición del evento y un retardo del evento máximo (K'_D) de la pluralidad de eventos operativos de entrada, en donde el valor del retardo de evento máximo (K'_D) está basado en la información transferida sobre los retardos de eventos y sobre la base de los retardos de eventos de la pluralidad de eventos operativos de entrada (e; A, B, C) recibidos desde los detectores de eventos de origen asociados.
13. Un programa informático que tiene un código de programa para realizar el método según una de las reivindicaciones precedentes, cuando el programa informático se ejecuta en un ordenador o procesador.
14. Un aparato para la migración de un detector de eventos de un primer nodo (N_1) de un sistema informático distribuido (200) hacia un segundo nodo (N_2) del sistema informático distribuido, en donde el detector de eventos procesa una pluralidad de eventos operativos de entrada (e, A, B, C) para generar al menos un evento operativo de salida (D) sobre la base de los eventos de entrada, en donde cada uno de los de eventos de entrada está asociado a un retardo de evento individual que corresponde a un tiempo de propagación del evento de entrada hacia el primer nodo (N_1), cuyo aparato comprende:
medios para la recepción, en el segundo nodo (N_2), de la pluralidad de eventos de entrada (e; A, B, C) reenviados desde el primer nodo (N_1) y, en paralelo, desde los detectores de eventos origen asociados con la pluralidad de eventos de entrada; y

medios para el procesamiento, en el segundo nodo (N_2), de la pluralidad de eventos operativos de entrada (e; A, B, C) recibidos sobre la base de la información transferida sobre los retardos de eventos y sobre la base de los retardos de eventos de la pluralidad de eventos operativos de entrada (e; A, B, C) recibidos desde los detectores de eventos de origen asociados, en donde el medio para el procesamiento está configurado para determinar un tiempo para retransmitir un evento de entrada recibido hacia el detector de eventos objeto de migración sobre la base de su tiempo de aparición del evento operativo y un retardo de evento máximo (K'_D) de la pluralidad de eventos operativos de entrada, en donde el valor de retardo de evento máximo (K'_D) está basado en la información transferida sobre los retardos de eventos y sobre la base de los retardos de eventos de la pluralidad de eventos de entrada (e; A, B, C) recibidos desde los detectores de eventos de origen asociados.

15. Un sistema informático distribuido (200) para efectuar la migración de un detector de eventos operativos de un primer nodo (N_1) de un sistema informático distribuido (200) a un segundo nodo (N_2) del sistema informático distribuido, en donde el detector de eventos procesa una pluralidad de eventos operativos de entrada (e; A, B, C) para generar al menos un evento de salida (D) basado en los eventos de entrada, en donde cada uno de los eventos operativos de entrada está asociado con un retardo de evento individual correspondiente a un tiempo de propagación del evento de entrada hacia el primer nodo (N_1), comprendiendo el sistema informático distribuido (200):

medios para transferir, desde el primer nodo (N_1) al segundo nodo (N_2) información sobre retardos de eventos correspondientes a la pluralidad de eventos operativos de entrada (e; A, B, C) en el primer nodo (N_1);

medios para transferir el contenido del detector de eventos desde el primer nodo (N_1) al segundo nodo (N_2) para obtener un detector de eventos objeto de migración en el segundo nodo;

medios para reenviar la pluralidad de eventos operativos de entrada (e; A, B, C) desde el primer nodo (N_1) al segundo nodo (N_2);

medios para la recepción, en el segundo nodo (N_2), de la pluralidad de eventos operativos de entrada (e; A, B, C) reenviados desde el primer nodo (N_1) y, en paralelo, desde los detectores de eventos de origen asociados a la pluralidad de eventos de entrada; y

medios para el procesamiento, en el segundo nodo (N_2), de la pluralidad de eventos operativos de entrada (e; A, B, C) recibidos sobre la base de la información transferida sobre los retardos de eventos y sobre la base de los retardos de eventos de la pluralidad de eventos operativos de entrada (e; A, B, C) recibidos desde los detectores de eventos de origen asociados, en donde los medios para el procesamiento están configurados para determinar un tiempo para retransmitir un evento de entrada recibido hacia el detector de eventos objeto de migración sobre la base de su tiempo de aparición del evento operativo y un retardo de evento máximo (K'_D) de la pluralidad de eventos operativos de entrada, en donde el valor de retardo de evento máximo (K'_D) está basado en la información transferida sobre los retardos de eventos y sobre la base de los retardos de eventos de la pluralidad de eventos de entrada (e; A, B, C) recibidos desde los detectores de eventos de origen asociados.

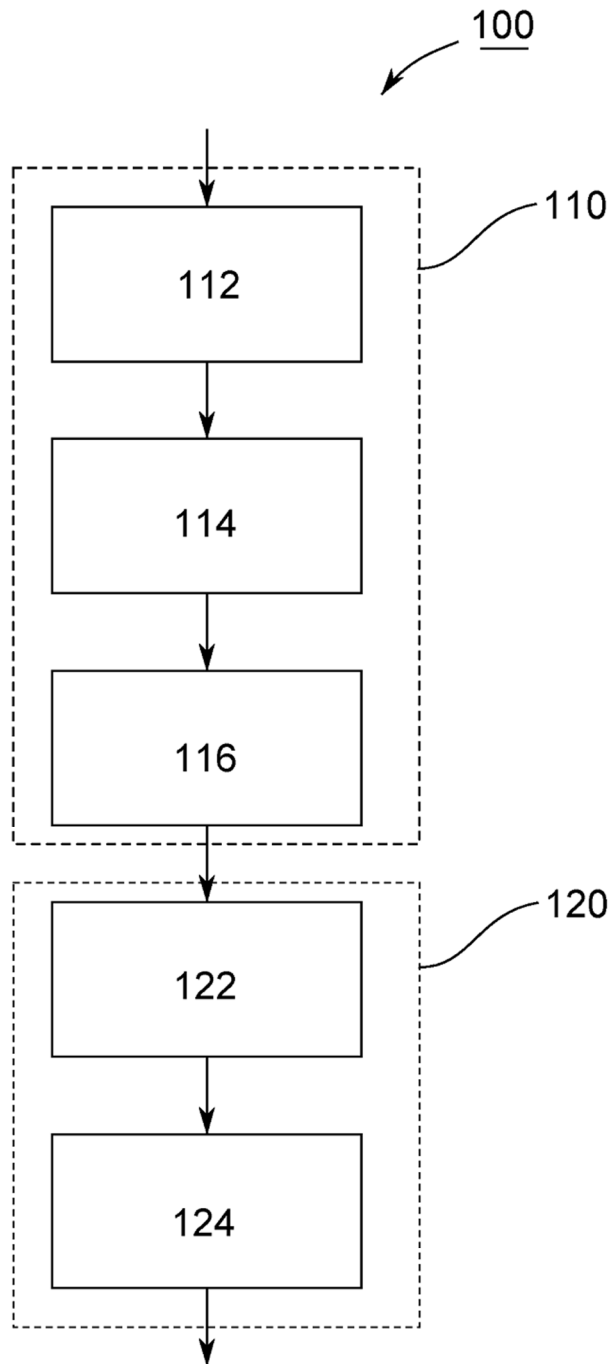


Fig. 1

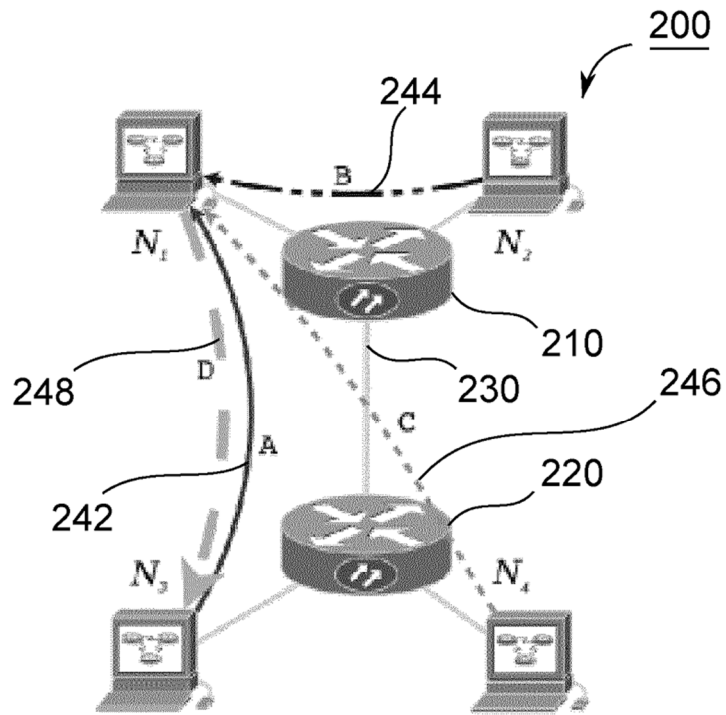


Fig. 2a

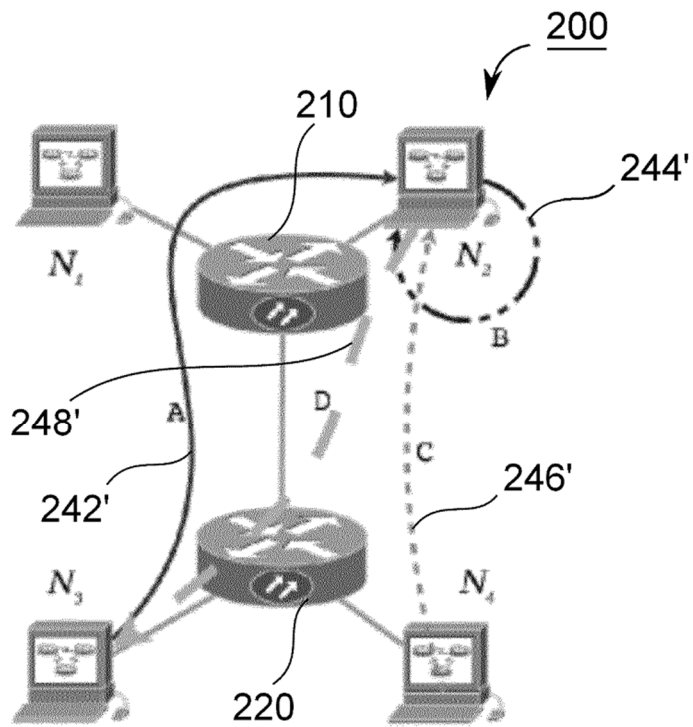


Fig. 2b

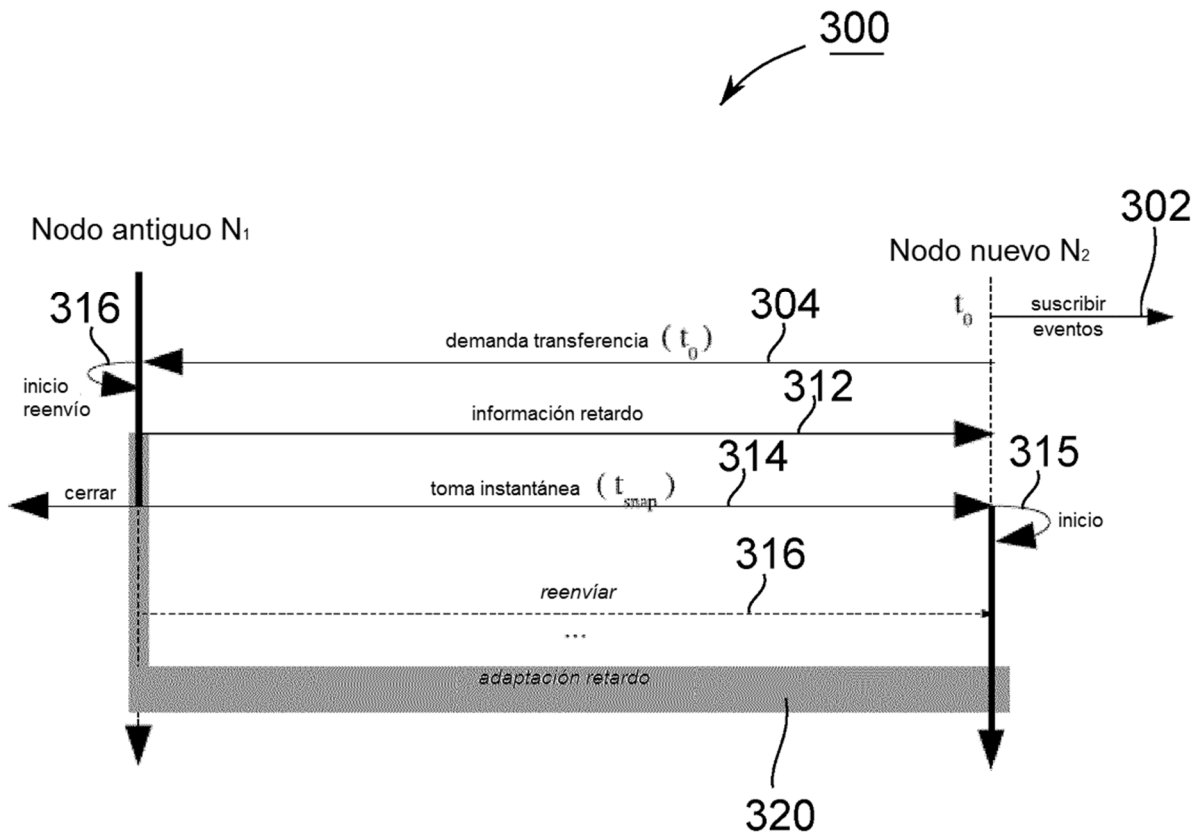


Fig. 3

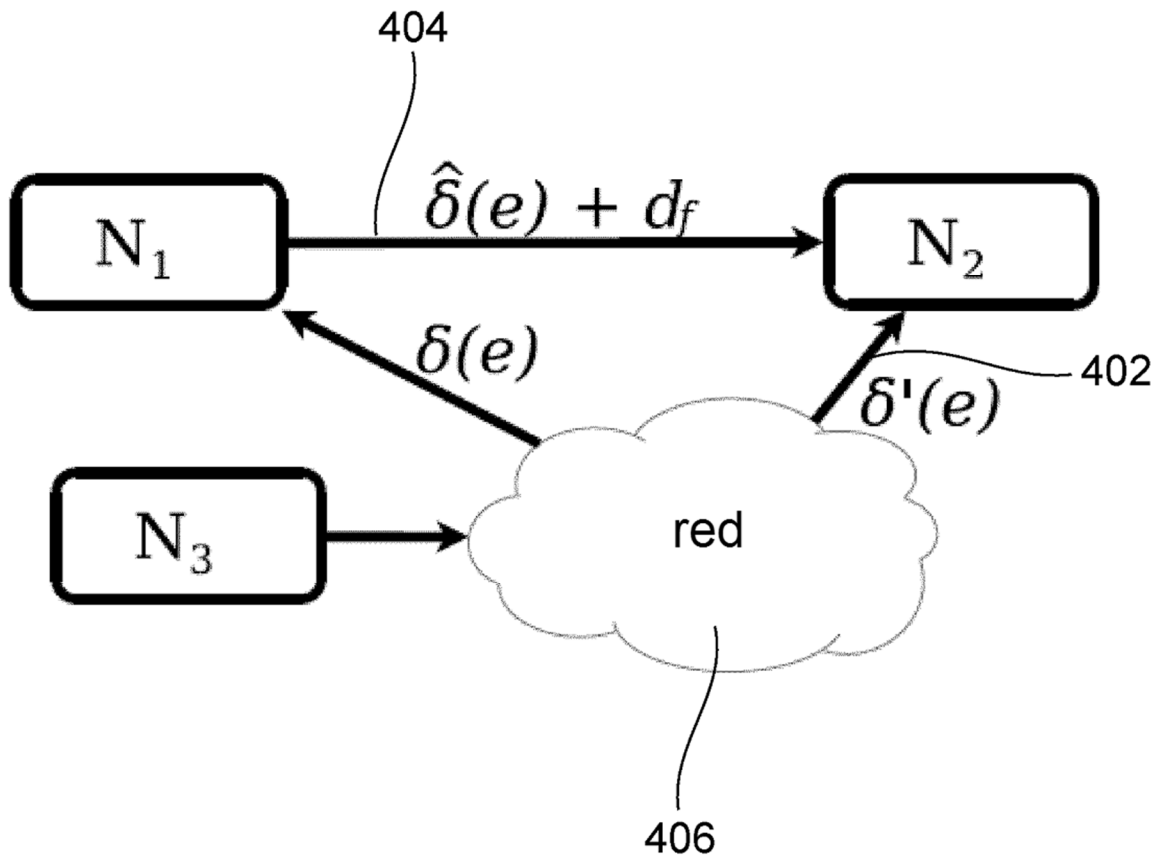


Fig. 4

 Algoritmo 3: Adaptación de retardos. Cancelación de ecos operativos

Datos: ED_{new} , d_f , Lista de retardos previstos

```

begin
502 →  $EventList$  eco;
      for retardo  $d$  : retardos do
504 →   |  $d \leftarrow d + d_f$ ;
506 →    $ED_{new}.K \leftarrow K \leftarrow \max(\text{retardos})$ ;           // establecer  $K$ 
       $Event$   $e$ ;
508 →   while  $e.recepción()$  do
          if  $e.id \notin ED_{new}.GetSubscriptions()$  then
510 →     | continuar:
512 →     if  $e.isDirect()$  then
514 →       |  $delays.at(e.id) \leftarrow clk - e.ts$ ;
516 →       |  $K_n \leftarrow \max(\text{retardos})$ ;
          if  $K_n \neq K$  then
518 →       |  $ED_{new}.K \leftarrow K \leftarrow K_n$ ;           // regenerar  $K$ 
          |  $NotifyDelayChange()$ ;
           $StopForward(e.id)$ ;
520 →     if  $echo.contiene(e)$  then
522 →       |  $echo.borrar(e)$ ;           // duplicar.
          else
524 →       |  $echo.añadir(e)$ ;
          |  $ED_{new}.pushToOrderingUnit(e)$ ;
  
```

Fig. 5