

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 431 931**

51 Int. Cl.:

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: **26.05.2009 E 09161071 (7)**

97 Fecha y número de publicación de la concesión europea: **31.07.2013 EP 2256632**

54 Título: **Planificación de multi-procesadores**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
28.11.2013

73 Titular/es:

**TELEFONAKTIEBOLAGET L M ERICSSON
(PUBL) (100.0%)
164 83 Stockholm, SE**

72 Inventor/es:

CARLSSON, ANDERS

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 431 931 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

Planificación de multi-procesadores

5 Campo técnico
La presente invención se refiere a la planificación de tareas de procesamiento en un sistema que tiene una pluralidad de unidades de procesamiento.

10 Antecedentes
La planificación de tareas de software ha sido extensamente investigada durante más de cuarenta años, y continúa siendo un tema de intensa investigación y desarrollo. Una razón es que nunca hay una planificación ideal. Sin embargo, la planificación de tareas en un solo procesador es muy bien conocida, y se ha probado que heurísticas bien conocidas resultan óptimas para tareas en tiempo real en un único procesador.

15 Las planificaciones de tareas sobre un grupo de procesadores están menos exploradas. La dimensión añadida de múltiples procesadores da lugar a varios planteamientos que gestionan la complejidad añadida. Ejemplos de planteamientos básicos incluyen:

20 *Planificación global, migración completa.* En este planteamiento hay una sola cola para todos los procesadores (planificación global) y migración completa, es decir los procesadores puede en cualquier momento interrumpir una tarea en favor de una tarea de mayor prioridad, dejando que la tarea interrumpida sea manejada más tarde por otro procesador.

25 *Planificación global, migración limitada.* En este planteamiento hay también una sola cola para todos los procesadores. Una vez que una tarea es asumida por un procesador, es completada sin interrupción por el mismo procesador.

Planificación local, particionada. En este planteamiento, hay una cola separada para cada procesador. Las tareas de un cierto tipo siempre llegan a una cierta cola.

30 *Planificación de dos niveles, o híbrida.* En este planteamiento, hay una cola global para todos los procesadores. Las tareas son distribuidas desde la cola global a colas locales para cada procesador.

Para sistemas reales distribuidos o de multi-núcleo, es también necesario considerar el impacto de los retardos de envío, el rendimiento del sistema de memoria y la estructura de interconexión. Éstos añaden impedimentos no triviales al rendimiento para un sistema de multi-procesadores.

35 Considerando la planificación global con migración completa, es bien conocido que el procesamiento de tareas en orden ascendente de plazos, "Plazo más Temprano Primero" (EDF – Earliest Deadline First, en inglés), no es óptimo para el multi-procesador. Existen conjuntos de tareas que se aproximan a una utilización global cero, mientras que todavía no es planificable mediante EDF. Además, el uso de EDF con migración completa puede provocar una media de una migración por tarea. Si el número de procesadores es grande, el número de invocaciones del planificador puede también colapsar a un único nodo, independientemente de que EDF sea un algoritmo simple.

40 Un algoritmo popular para la planificación de multi-núcleo global es "El de Menor Relajación Primero" (LLF – Least Laxity First, en inglés), que tiene mejores propiedades que el EDF. Por su parte, es capaz de utilizar la mitad de la capacidad agregada para tareas en tiempo real. No obstante, como el EDF, el LLF también provoca una gran cantidad de migraciones y de invocaciones del planificador. Además, la complejidad de cálculo del LLF es ligeramente mayor que la del planteamiento del EDF.

45 Para poder aumentar más la utilización disponible para procesamiento en tiempo real, es factible utilizar uno de los planificadores de multi-núcleo óptimos, bien el algoritmo de "Imparcialidad Proporcional" (PFair - Proportionate Fair, en inglés) o "Tiempo de Ejecución restante Local más Largo Primero" (LLREF – Largest Local Remaining Execution time First, en inglés). El algoritmo PFair, no obstante, provoca una cantidad de migraciones incluso mayor que el EDF. Aunque LLREF produce una menor cantidad de invocaciones del planificador y menos migraciones, el LLREF necesita conocer cuándo será liberada la siguiente tarea para ser liberada. Por esta razón el LLREF puede ser considerado un algoritmo que no es muy adecuado para el procesamiento en tiempo real.

50 Los algoritmos de planificación global como FP, EDF, LLF, PFair, etc., pueden ser académicamente interesantes, pero fallan cuando son aplicados a un sistema real con retardos y un rendimiento limitado del sistema de memoria. Con una carga cada vez mayor, estos planificadores provocan una cantidad de migraciones de tarea cada vez mayor. Puesto que las migraciones de tarea implican un aumento de carga, la frecuencia de migración también aumentará. Bajo ciertas condiciones, particularmente cerca de algún límite de utilización práctico, el aumento de carga de las migraciones se acelera, conduciendo a una carga inestable que rápidamente inutiliza todo el sistema.

60 Un ejemplo de planificación de la técnica anterior en un sistema de procesador de multi-núcleo se describe en la publicación de la solicitud de patente de US 2007/0220517. No obstante, la planificación descrita en esta memoria

está asociada con los respectivos inconvenientes de las altas velocidades de migración y, en caso de utilizar EDF, una baja utilización en tiempo real. El modo de planificación de histéresis descrito en el documento de US 2007/0220517 puede mejorar el resultado, pero también aumenta los tiempos de respuesta, lo que limita su utilidad para aplicaciones en tiempo real duras.

5 El documento que sigue describe una estructura de planificador en tiempo real de dos niveles (Calidad de Servicio/QoS – Quality of Service, en inglés) en multi-grupos y mallas.

10 LIGANG HE ET AL: “Dynamic Scheduling of Parallel Jobs with QoS Demands in Multiclusters and Grids” GRID COMPUTING, 2004. PROCEEDINGS. FIFTH IEEE/ACM INTERNATIONAL WORKSHOP ON PITTSBURGH, PA, USA 08-08 NOV. 2004, PISCATAWAY, NJ, USA. IEEE, 8 de Noviembre de 2004 (2004-11-08), páginas 402-409, XPO10769526.

15 El aspecto particular de la rellamada y la redistribución de tareas puestas en cola para procesadores ocupados a procesadores en reposo es una estrategia bien conocida (“robo de tarea/trabajo”), como puede encontrarse por ejemplo en el siguiente documento:

20 YUE, J.: “Global Backfilling Scheduling in Multiclusters” LECTURE NOTES IN COMPUTER SCIENCE LNCS3285, 2004, páginas 232-239, XP002541638

Compendio

La invención se define en las reivindicaciones 1 y 2 independientes del método, y de manera correspondiente en la reivindicación 17 del dispositivo y en una correspondiente reivindicación 18 para un programa de ordenador.

25 Por ello, para mejorar las soluciones de la técnica anterior se proporciona, de acuerdo con un primer aspecto, un método de planificar tareas de procesamiento en un aparato de cálculo de multi-procesadores. Las tareas están asociadas con al menos un plazo respectivo. El método implica la comunicación entre una función de planificación de tareas global y una pluralidad de funciones de planificación de tareas local, estando cada función de planificación de tareas local asociada con un procesador respectivo. El método comprende guardar, en la función de planificación global, información del estatus actualizada relativa a cualquier procesador en reposo, a momentos del tiempo en los que se espera que cualquier procesador se encuentre en reposo y a valores de relajación asociados con cualquier procesador. Además, el método comprende planificar, en la función de planificación de tareas global, tareas mediante el envío de tareas a las funciones de planificación de tareas locales y el envío está condicionado por la información de estatus. Las tareas son llamadas de las funciones de planificación de tareas locales y cualquier procesador que no esté en reposo y cualquier tarea llamada es enviada a una función de planificación de tareas local en un procesador en reposo. Las tareas interrumpidas son recibidas desde las funciones de planificación de tareas locales, y cualquier tarea interrumpida recibida es enviada a una función de planificación de tareas local, estando el envío condicionado por la información de estatus.

40 En otras palabras, las tareas son interrumpidas y devueltas a la función de planificación de tareas global si un procesador no puede terminar la tarea, los procesadores siguen trabajando en la tarea actual durante el mayor tiempo posible, hasta que es necesario cambiar para cumplir el plazo de otra tarea, y los procesadores que pasan a reposo pueden empezar a trabajar en tareas de la cola en otros procesadores. Este planteamiento es una combinación de conceptos. La planificación de lista global en la que una función de planificación de tareas global asigna tareas al nodo “menos ocupado”; una planificación local con política de “ejecutar tarea actual hasta que se complete”. La tarea ejecutada actualmente es interrumpida sólo si es necesario cumplir otros plazos; y el “robo de tarea” global, es decir, si cuando un procesador pasa a reposo hay más tareas que procesadores ocupados, se mueve una tarea desde el procesador más ocupado con más de una tarea al procesador que está en reposo.

50 El emisor global tiene baja complejidad y es básicamente invocado sólo cuando es necesario, es decir, cuando llegan nuevas tareas de procesamiento. Esto mejora la escalabilidad, de manera que matrices de procesadores más grandes puedan ser planificadas globalmente.

55 Se ha encontrado que, aplicando tal planificación en un entorno de multi-procesadores, el número de migraciones de tarea total es 5-10 veces menos que cuando se aplican planificación de EDF global y de LLF global. Esto conduce, por ejemplo, a una mejor utilización frente a largos retardos de bus.

60 El hecho de que cada procesador esté asociado con su propio planificador local permite una combinación de planteamientos global y particionado. El planificador local reporta su compromiso actual a la función de planificación de tareas global, que entonces automáticamente puede decidir si asignar tareas o no.

Se estima que la máxima utilización del tiempo real posible de una matriz de procesadores puede elevarse desde ligeramente por encima de la mitad de la capacidad hasta cerca de la capacidad total. Esto a su vez puede permitir una considerable reducción de la capacidad instalada en los sistemas de procesamiento en los cuales se realiza la

planificación, tales como sistemas de plataforma de comunicación mediante telefonía móvil, módems de telefonía móvil y estaciones de base en sistemas de comunicación mediante telefonía móvil, lo que lleva a reducir los costes de fabricación, alternativamente, conduciendo a una mayor vida útil a un menor coste.

5 Las realizaciones incluyen aquéllas en las que el mantener la información del estatus actualizada comprende recibir, desde cada función de planificación de tareas local, un valor de relajación de planificación calculado respectivo que supone una diferencia mínima entre los plazos de finalización para las tareas y los tiempos de finalización de tarea esperados asociados con tareas actualmente enviadas a cada procesador respectivo, y donde la planificación de una tarea comprende comprobar si existe un procesador en reposo. Si existe un procesador en reposo, la tarea es
10 enviada a la función de planificación de tareas local del procesador en reposo, o si no se calcula un tiempo de finalización esperado de la tarea, estando el cálculo relacionado con una situación que implica la existencia de un procesador en reposo, se calcula un valor de relajación de tarea como una diferencia entre el plazo para la tarea y el tiempo de finalización esperado calculado para la tarea, se realiza una comprobación acerca de si la relajación de la
15 tarea calculada es mayor que una diferencia entre el momento del tiempo más temprano en el que se espera que cualquier procesador esté en reposo y el tiempo actual. Si es así, la tarea es enviada a la función de planificación de tareas local de ese procesador, o si no la tarea es enviada a la función de planificación de tareas local del procesador que tiene el valor de relajación de planificación más grande.

20 Otras realizaciones incluyen aquéllas en las que mantener la información de estatus actualizada comprende recibir, desde cada función de planificación de tareas local, un valor de relajación de planificación calculado respectivo que es una diferencia mínima entre los plazos de finalización para tareas y los tiempos de finalización de tarea esperados asociados con las tareas actualmente enviadas a cada procesador respectivo, y donde la planificación de una tarea comprende comprobar si existe un procesador en reposo. Si existe un procesador en reposo, la tarea es
25 enviada a la función de planificación de tareas local del procesador en reposo, o si no se realiza una comprobación acerca de si un plazo de inicio para la tarea es posterior al momento más temprano del tiempo en el que se espera que cualquier procesador esté en reposo y, si es así, la tarea es enviada a la función de planificación de tareas local de ese procesador. Si un plazo de inicio para la tarea no es posterior al momento más temprano del tiempo en el que se espera que cualquier procesador esté en reposo, la tarea es enviada a la función de planificación de tareas local del procesador que tiene el valor de relajación de planificación más grande.

30 Esto es, en lugar de un tiempo de finalización, las tareas pueden tener un plazo de inicio asociado, que puede ser comparado directamente con los tiempos de finalización de planificación de procesadores sin el cálculo intermedio de la relajación, lo que puede simplificar la implementación.

35 Las realizaciones incluyen aquéllas que comprenden, en la función de planificación de tareas global, determinar a partir de la información de estatus actualizada si al menos un procesador está en reposo y si es así, determinar los datos de decisión de llamada que comprenden información relativa al número total de tareas que actualmente son
40 enviadas a las funciones de planificación de tareas locales, el número de procesadores que no se encuentran en reposo e información relativa a cuál de los procesadores es el más ocupado. A continuación se decide, utilizando los datos de decisión de llamada, si llamar a una tarea desde el procesador más ocupado y, si es así, se envía una solicitud de llamada al procesador más ocupado, se recibe una tarea desde el procesador más ocupado y la tarea recibida es enviada a la función de planificación de tareas local de un procesador en reposo.

45 En tales realizaciones que implican solicitudes de llamada, una función de planificación de tareas local puede llevar a cabo la recepción de una solicitud de llamada de tarea desde la función de planificación de tareas global, la estimación de si las tareas de la cola pueden cumplir sus respectivos plazos de finalización si fuesen procesadas por cualquier procesador y, dependiendo de la estimación, la devolución de una tarea a la función de planificación de tareas global.

50 La llamada de tareas, o "robo de tareas", hace que esta planificación de dos niveles combinados suponga un ahorro de trabajo. Esto es, siempre que haya al menos tantas tareas como procesadores, ningún procesador se deja en reposo, lo que lleva a menores longitudes de planificación, lo cual también aumenta la utilización posible. El robo de
55 tarea puede ser considerado como la corrección de una decisión por parte de la función de planificación de tareas global.

60 Además, en tales realizaciones, una función de planificación de tareas local puede realizar una determinación de criterios de selección de devolución para decidir qué tarea devolver a la función de planificación de tareas global. La determinación de criterios de selección de devolución puede comprender cualquiera de una tarea última en una cola de tareas, calcular valores de retardo de reenvío para tareas de la cola y comparar los valores del retardo de reenvío con valores de relajación de las tareas de la cola, calcular los valores del retardo de reenvío para tareas de la cola y determinar un tiempo de inicio respectivo, teniendo en cuenta los valores del retardo de reenvío, en cualquier procesador y calcular estos valores de inicio, y a continuación la determinación de los criterios de selección de devolución devolviendo una tarea a la función de planificación de tareas global de acuerdo con los valores de selección de devolución.

Esto asegura que las tareas son devueltas sólo si se estima que pueden cumplir sus plazos, teniendo en cuenta los retardos entre decisiones por parte de la función de planificación de tareas global y la reacción a las decisiones por parte de las funciones de planificación de tareas locales, y viceversa.

5 Las realizaciones incluyen aquéllas que comprenden, en una función de planificación de tareas local, comprobar si la relajación de planificación es menor de cero y, si es así, calcular un momento del tiempo de expiración que es el momento del tiempo en el cual una tarea de ejecución actualmente debe dejar de ejecutarse con el fin de que la relajación de planificación sea cero, interrumpir la tarea que se está ejecutando actualmente, teniendo lugar la interrupción en un intervalo de tiempo distinto de cero antes del momento del tiempo de expiración, devolver la tarea interrumpida a la función de planificación de tareas global, activar una tarea de la cola y recalcular la relajación de planificación.

10 En otras palabras, si la relajación de planificación resulta ser negativa, la función de planificación de tareas local establece que un primer temporizador expire lo más tarde posible cuando sea necesario iniciar la ejecución de la primera tarea de la cola para que la relajación de la planificación se convierta en cero. Cuando el temporizador expira, la tarea actual es devuelta a la función de planificación de tareas global y la primera tarea de la cola es activada.

15 Permitiendo que los procesadores con exceso de tareas se desconecten, es decir, se interrumpan, una tarea habilita la compartición de carga sobre el grupo de procesadores, aumentando la posible utilización para el procesamiento en tiempo real. Esto resuelve también el problema de la separación de tareas, un elemento clave de los algoritmos de planificación en tiempo real óptimos.

20 En realizaciones tales que implican relajación de planificación negativa, el intervalo de tiempo distinto de cero puede ser determinado calculando la relajación de la tarea que se está ejecutando actualmente, calculando una relajación de asunción para cada tarea de la cola, como la diferencia entre un tiempo de finalización esperado y un tiempo de finalización con respecto a un momento de inicio asumido común, e identificando la menor relajación de asunción de entre las tareas de la cola. El intervalo de tiempo distinto de cero es ajustado, en caso de que la relajación de la tarea que se está ejecutando actualmente sea mayor que la menor relajación de asunción identificada, a un primer valor de intervalo del temporizador, y si no, el intervalo de tiempo distinto de cero es ajustado a un segundo valor del intervalo del temporizador, siendo el segundo valor del intervalo del temporizador mayor que el primer valor del intervalo del temporizador.

25 Alternativamente, en realizaciones que implican relajación de planificación negativa, el intervalo de tiempo distinto de cero puede ser determinado calculando la relajación de la tarea que se está ejecutando actualmente, calculando una relajación de asunción para cada tarea de la cola, como la diferencia entre un tiempo de finalización esperado y un tiempo de finalización con respecto a un momento del tiempo de inicio asumido común, identificando la menor relajación de asunción de entre las tareas de la cola. El intervalo de tiempo distinto de cero es puesto a un valor que es proporcional a la diferencia entre la relajación de la tarea que se está ejecutando actualmente y la menor relajación de asunción e inversamente proporcional a la suma de la relajación de la tarea que se está ejecutando actualmente y la menor relajación de asunción.

30 Esto proporciona un efecto ventajoso porque una tarea actual con relajación grande es suspendida lo antes posible si una tarea con una relajación considerablemente menor está en la cola, mientras que al mismo tiempo se tienen en cuenta el hecho de que resulta deseable que una tarea actual con menor relajación progrese lo más posible antes de ser interrumpida.

35 Además, tales realizaciones mejoran la capacidad de contrarrestar el hecho de que la expiración del temporizador debe ocurrir de manera que al menos se haga algún progreso en la tarea actual antes de que sea devuelta a la función de planificación de tareas global. Permitir una inmediata expiración puede conducir a que las tareas sean rotadas infinitamente entre una tarea función de planificación de tareas local y la función de planificación de tareas global sin que nunca se trabaje en ellas, eventualmente sin que se cumpla su plazo. En otras palabras, tales realizaciones mejoran la capacidad de planificar cargas de trabajo importantes sobre las reglas de un solo valor típicamente utilizadas en las soluciones de la técnica anterior.

40 Además, las realizaciones que implican relajación de planificación negativa pueden comprender, en una función de planificación de tareas local, recibir una solicitud de llamada de tarea desde la función de planificación de tareas global, estimar si las tareas de la cola pueden cumplir sus respectivos plazos de finalización teniendo en cuenta cualquier retardo provocado por el guardado y la restauración de datos asociados con la replanificación entre procesadores y, dependiendo de la estimación, devolver una tarea a la función de planificación de tareas global.

45 Tales realizaciones tratan problemas relacionados con los retardos implicados durante la migración de tarea. Esto es, la en ocasiones significativa cantidad de tiempo (miles de ciclos de reloj) empleada cuando se guarda el estado

de una tarea de un procesador a un área de memoria común, seguida por la recarga de otro procesador de la misma área de memoria común.

5 Las realizaciones incluyen aquéllas que comprenden, en una función de planificación de tareas local, comprobar si la relajación de planificación es menor de cero y, si es así, calcular un momento del tiempo de expiración, siendo el momento del tiempo el momento del tiempo en el cual una tarea que se está ejecutando actualmente debe dejar de ejecutarse con el fin de que la relajación de planificación sea cero, calcular una relajación de planificación de asunción dada la asunción de que la tarea que se está ejecutando actualmente está replanificada. Si la diferencia calculada es positiva, la tarea que se está ejecutando actualmente es interrumpida en un intervalo de tiempo distinto de cero antes del momento del tiempo de expiración, y si no la tarea que se está ejecutando actualmente es interrumpida de manera inmediata. Una tarea de la cola es a continuación activada y la relajación de planificación es recalculada.

15 Tales realizaciones tratan con la situación que implica a tareas que están estáticamente asignadas a un procesador. Tales tareas estáticas resultan parte del compromiso de un procesador y son utilizadas en el cálculo del periodo de ocupación y de la relajación. No obstante, puesto que son asignadas estáticamente no pueden ser pasadas a la función de planificación de tareas global cuando son interrumpidas y no pueden ser transferidas a la función de planificación de tareas global si otro procesador está en reposo.

20 Las realizaciones incluyen aquéllas que comprenden, en una función de planificación de tareas local, asignar un nivel de prioridad a cada tarea, manteniendo una pluralidad de colas de tareas, conteniendo cada cola tareas que tienen un único nivel de prioridad, y manteniendo, en la función de planificación de tareas global, información de estatus actualizada relativa a la pluralidad de colas.

25 Añadiendo el concepto de prioridad, tales realizaciones serán capaces de mejorar la capacidad del proceso de planificación para diferenciar entre tareas en términos de qué tareas son más críticas, por ejemplo, en términos de que no se cumpla un plazo, y qué tareas son menos críticas.

30 Las realizaciones incluyen aquéllas en las que la función de planificación de tareas global está asociada con un procesador específico y aquéllas en las que la función de planificación de tareas global está asociada con una pluralidad de procesadores.

Tales realizaciones permitirán una más flexible implementación de la planificación.

35 Las realizaciones incluyen aquéllas en las que el mantenimiento de información de estatus actualizada comprende mantener una lista de procesadores clasificada en orden de momentos del tiempo ascendentes en los que se espera que los procesadores estén en reposo, y mantener una lista de procesadores clasificados en orden de relajación descendente. El mantenimiento de información de estatus actualizada puede comprender también mantener una lista de primero que entra primero que sale de procesadores en reposo.

40 Tales realizaciones permitirán una implementación más simplificada de la planificación.

45 En un segundo aspecto, se proporciona un programa de ordenador que comprende instrucciones de software que, cuando son ejecutadas en un dispositivo de procesamiento, llevan a cabo el método de acuerdo con el primer aspecto.

50 En un tercer aspecto se proporciona un dispositivo de procesamiento que comprende medios de procesamiento y de memoria configurados para llevar a cabo el método tal como se ha resumido anteriormente, y en un cuarto aspecto se proporciona un dispositivo de comunicación que comprende tal dispositivo de procesamiento.

Estos otros aspectos proporcionan efectos y ventajas correspondientes tal como se ha explicado anteriormente junto con el primer aspecto.

55 Breve descripción de los dibujos

Se describirán ahora realizaciones con referencia a los dibujos que se acompañan, en los cuales:

60 la figura 1 ilustra esquemáticamente entidades de planificación que interactúan,
la figura 2 ilustra esquemáticamente entidades de planificación que interactúan,
la figura 3 ilustra esquemáticamente una plataforma de procesamiento,
las figuras 4a y 4b son diagramas de flujo de planificación,
la figura 5 es una planificación de procesador,
la figura 6 es una planificación de procesador,
las figuras 7a a 7c son diagramas de flujo de planificación, y
las figuras 8a a 8e son diagramas de flujo de planificación.

Descripción detallada de realizaciones

5 La figura 1 ilustra esquemáticamente un sistema 100 de procesamiento que comprende un número de entidades que interactúan. Las entidades se describen en términos funcionales y comprenden tanto unidades de hardware como unidades de software. Como resultará evidente para un experto en la materia, las implementaciones del sistema 100 pueden diferir en términos de qué y cuánta funcionalidad de cada entidad está realizada mediante hardware, así como qué y cuánta funcionalidad está realizada en software. Una implementación de ejemplo se describirá brevemente a continuación junto con la figura 3.

10 Estas unidades de procesamiento 108a-c, por ejemplo procesadores individuales o núcleos de procesador, que tienen cada uno una memoria local 110a-c respectiva, están conectados a un bus 106 de interconexión. Una memoria global 104 está también conectada al bus 106. Cada unidad de procesamiento 108a-c está en conexión con una función de planificación de tareas local 112a-c respectiva. Una función de planificación de tareas global 102 está en conexión con las funciones de planificación de tareas locales 112-a-c y con la memoria global 104 a través del bus 106. La función de planificación de tareas global 102 puede estar realizada mediante software que se ejecuta en una cualquiera o en más de las unidades de procesamiento 108a-c. Esto es, la función de planificación de tareas global 102 puede ser centralizada así como descentralizada.

20 Como se describirá con detalle a continuación, la función de planificación de tareas global 102 y las funciones de planificación de tareas locales 112a-c se comunican entre sí por medio de algoritmos que utilizan datos almacenados en las diferentes unidades de memoria 104, 110a-c. En lo que sigue, se asume que las tareas de procesamiento son entidades que están definidas como conjuntos de instrucciones y parámetros que son creados por aplicaciones de software o por un sistema operativo que se ejecuta en uno o más de los procesadores 108a-c.

25 Los datos y parámetros que se explicarán en lo que sigue incluyen listas y colas. Éstas están almacenadas, como resultará evidente para un experto en la materia, en las unidades de memoria 104, 110a-c de manera apropiadas de acuerdo con cada situación específica. Por ejemplo, la función de planificación de tareas global 102 se refiere a listas de procesadores y a una cola de envío global. Éstas están típicamente guardadas en la memoria global 104. Las funciones de planificación de tareas locales 110a-c tienen colas de envío locales que típicamente son guardadas en las memorias locales 110a-c.

35 La figura 2 ilustra un sistema 200 similar al sistema 100 de la figura 1. No obstante, en lugar de tener un bus (106 en la figura 1) de interconexión, el sistema 200 es de un tipo que tiene una topología de bus de anillo. Específicamente, esto conlleva que las unidades de procesamiento 208a-c, cada una con memoria local 210a-c y respectivas funciones de planificación de tareas locales 212a-c, estén conectadas en forma de bus 206 de anillo. Una función de planificación de tareas global 202 con una memoria local 216 está en comunicación con las funciones de planificación de tareas locales 212a-c. Esto es para ilustrar que la planificación tal como se describe en esta memoria puede ser llevada a cabo también en una topología en la cual los retardos en la comunicación entre la función de planificación de tareas global y las funciones de planificación de tareas locales son muy diferentes de los retardos de comunicación experimentados en una topología de bus tal como la de la figura 1.

40 La figura 3 es un diagrama de bloques funcional que ilustra esquemáticamente una plataforma de procesamiento que forma en todo o en parte una disposición 300 de comunicación mediante telefonía móvil. La disposición 300 comprende un procesador 301, una memoria 303, circuitos de frecuencia de radio 307 y una unidad de I/O (Entrada/Salida – Input/Output, en inglés) 309 que representa una colección de unidades de entrada/salida tales como un visualizador, un teclado numérico, un micrófono y un altavoz.

50 El sistema 100 de la figura 1 y el sistema 200 de la figura 2 pueden ser implementados en la disposición 300. En tales realizaciones, las funciones de planificación de tareas global 102, 202, las funciones de planificación de tareas locales 112, 212 y la memoria local 110, 210, 216 están todas formando una parte 305 de la memoria 303 y las unidades de procesamiento 108, 208 pueden ser en forma de diferentes núcleos de procesador de la unidad de procesamiento 301.

55 En referencia ahora también a las figuras 4 a 6, las tareas de planificación y de procesamiento se describirán por medio de diagramas de flujo o algoritmos y de correspondientes ilustraciones de planificaciones de procesador. Las entidades implicadas en la planificación para ser descritas pueden ser las puestas como ejemplo anteriormente junto con las figuras 1 y 2, e implementadas en una disposición tal como la plataforma de procesamiento 300 de la figura 3. Debe observarse, no obstante, que la planificación puede ser realizada con cualquier sistema adecuado, con o sin memoria compartida, y en cualquier topología tal como anillo, malla, conmutador y en definitiva, en cualquier topología de compartición de datos tal como SMMP, NWOP, NUMA, y oculta. La planificación puede ser llevada a cabo en cualquier sistema que tenga cualquier número de migraciones, memoria común, bus compartido y sistemas que tengan un emisor de tareas separado como función de planificación de tareas global.

Desde un punto de vista global, las tareas son asignadas al procesador menos ocupado. La función de planificación de tareas global 102, 202 puede mantener tres listas de nodos planificables, es decir, procesadores: una primera lista en forma de una lista de Primero en Entrar Primero en Salir (FIFO – First In First Out, en inglés) de procesadores en reposo, una segunda lista en forma de una lista de procesadores clasificados en orden ascendente de tiempo de finalización de planificación. El tiempo de finalización de planificación es el tiempo en el que se espera que el procesador pase a reposo. Una tercera lista es en forma de una lista de procesadores clasificados en orden descendente de relajación de planificación. La relajación de planificación es el mínimo de la diferencia entre plazos de tarea y tiempos de finalización de tarea esperados, tomada sobre todas las tareas actualmente asignadas al procesador.

Una implementación equivalente, menos compleja, es utilizar dos listas de procesadores planificables en lugar de tres listas. Esto es, una primera lista de procesadores en orden ascendente de tiempo de finalización y una segunda lista de procesadores en orden descendente de relajación. El tiempo de finalización de procesadores en reposo estará entonces en el pasado. La relajación de un procesador en reposo sería infinita. La ordenación de procesadores en una implementación tal se hará automáticamente igual al orden de FIFO en las implementaciones que utilizan tres listas. No obstante, cuando el tiempo de finalización de un procesador es en un tiempo pasado, debe utilizarse el tiempo actual cuando se estima el tiempo de finalización de una tarea en un procesador en reposo.

Con referencia específica al diagrama de flujo de la figura 4A, se describirá la planificación mediante la función de planificación de tareas global 102, 202.

Cuando se recibe una nueva tarea en una etapa 401 de recepción, la tarea se sitúa en la cola de envío. Durante esta etapa 401 la relajación de planificación y los tiempos de finalización actualizados, recibidos desde las funciones de planificación de tareas locales 112, 212, están disponibles para la función de planificación de tareas global 102, 202. A la llegada de una nueva tarea, se realiza una comprobación acerca de si las tareas de la cola pueden ser enviadas o no. La condición para ello es una de lo siguiente: la lista de procesadores en reposo no está vacía, las listas de procesadores en tiempo de finalización ascendente y de relajación descendente contienen información válida para una mayoría de procesadores. Se toma una decisión en la etapa 403 de decisión de manera que, si no se cumple ninguna de las condiciones anteriores, la función de planificación de tareas global se detiene y espera a las actualizaciones de estatus de las funciones de planificación de tareas locales 112, 212, o recibe más tareas.

Si pueden enviarse tareas, la función de planificación de tareas global 102, 202 intenta primero asignar la tarea a un procesador en reposo. Esto se lleva a cabo por medio de una etapa 405 de comprobación. Si existe un procesador en reposo, la tarea es asignada a este procesador en una etapa 407 de asignación y el flujo continúa en una etapa 417 de reconocimiento de recepción, como se recibirá con más detalle en lo que sigue.

Si se encuentra en la etapa 405 de comprobación que no hay ningún procesador en reposo, la función de planificación de tareas global 102, 202 calcula, en una etapa 409 de cálculo, el tiempo de finalización esperado de la tarea en cualquiera de los procesadores 108, 208, si hubiera habido un procesador en reposo, y la diferencia entre el plazo de la tarea y el tiempo de finalización esperado, es decir, la máxima relajación de tarea. A continuación, como se decidió en una etapa 411 de decisión, si la relajación de la tarea es mayor que la diferencia entre el tiempo de finalización del primer tiempo de finalización y el tiempo actual, la tarea es asignada a ese procesador en la etapa 413 de asignación. Si no, la tarea es asignada y enviada al procesador con la mayor relajación de planificación en una etapa 415 de asignación.

El procesador reconoce la asignación, junto con la relajación de planificación actualizada y los tiempos de finalización, como se ilustra mediante la etapa 417 de reconocimiento. El procesador es a continuación eliminado de las listas en una etapa de eliminación 419 hasta que se recibe una etapa de actualización de estatus, como se ilustra mediante una devolución del flujo a la etapa 401 de recepción.

Volviendo ahora al diagrama de flujo de la figura 4B, se describirá el manejo de las llamadas a tareas. Cuando un procesador 108, 208 reporta un estatus que indica que el procesador se ha puesto en reposo, la función de planificación de tareas global 102, 202 sitúa el procesador en la lista de procesadores en reposo, como se ilustra mediante la etapa 451 de recepción de estatus y una etapa 453 de comprobación de reposo. Si hay tareas en la cola de envíos, éstas son enviadas, como se ilustra mediante la etapa 455 de tareas de comprobación, en las etapas 405 a 419.

Si no, si no queda ninguna tarea en la cola de envíos, si hay más tareas enviadas que procesadores actualmente activos, según se comprobó en una etapa 457 de comprobación, el emisor transmite un mensaje de llamada de tarea al procesador más ocupado con al menos una tarea en la cola, es decir, una tarea se ejecuta actualmente y al menos una tarea está en espera, como se ilustra mediante la etapa 459 de llamar a tareas. Si, en la etapa 457 de comprobación no se encuentra que haya más tareas enviadas que procesadores actualmente activos, el flujo vuelve a la etapa 451 de recibir estatus.

Se espera que un procesador responda a un mensaje de llamada devolviendo una tarea al emisor, como se indica mediante una etapa 461 de recibir tarea, el cual a su vez enviará la tarea recibida al procesador en reposo tal como se describe en las etapas 405 a 407. Qué tarea devolver, de las diferentes de la cola que hay en el procesador, se decide mediante la función de planificación de tareas local en el procesador, pero se espera que la tarea devuelta tenga suficiente relajación restante para que los retardos de envío no provoquen una pérdida de plazo.

Una variación de la planificación llevada a cabo por la función de planificación de tareas global tal como se describió anteriormente en una en la que, en lugar de utilizar plazos de finalización asociados con tareas, se utiliza un plazo de inicio para las tareas. Una característica de utilizar un plazo de inicio es que un plazo de inicio puede ser comparado directamente con los tiempos de finalización de planificación de procesadores sin el cálculo de relajación intermedio. Esto permite una implementación menos compleja, es decir, una implementación menos compleja de al menos la etapa 409 de cálculo.

Aunque la realización descrita anteriormente puede asumir implícitamente que una sola entidad física gestione la función de planificación de tareas global, la operación de la función de planificación de tareas global puede ser también implementada de tal manera que todos los participantes de un grupo de procesadores compartan la carga de trabajo de realizar la función de planificación de tareas global. En tales realizaciones, las (dos o tres) listas de información y la cola de envío están situadas en memoria, que es compartida, o al menos está accesible, para todos los procesadores. Cuando una tarea es emitida desde un procesador, por ejemplo, una tarea que pertenece a una tarea del sistema operativo, el procesador intenta enviar la tarea él mismo, bloqueando primero la cola de envío, insertando la tarea en la cola de envío y bloqueando las listas. A continuación la comprobación de condiciones de envío, el envío y la eliminación del objetivo del envío de las listas es llevado a cabo de la manera descrita anteriormente, junto con la figura 4A y la figura 4B. Esto es seguido a continuación por el desbloqueo de las listas y el desbloqueo de la cola de envío.

Aunque la descripción anterior se ocupa principalmente de la función de planificación de tareas global, la descripción a seguir se centrará principalmente en el procesamiento llevado a cabo por los procesadores 108, 208, tal como están controlados por sus respectivas funciones de planificación de tareas locales 112, 212. La función de planificación de tareas local de cada procesador controla el orden de ejecución básico de las tareas asignadas de acuerdo con cualquier algoritmo de planificación de un solo procesador adecuado. Cuando una tarea es asignada al procesador, es colocada en una cola de tareas local. Si el procesador estuviese en reposo inmediatamente antes de la recepción de la tarea, la tarea asignada se activa, es decir, se elimina de la cola local para la posición de tarea actual.

La función de planificación de tareas local 112, 212 calcula a continuación los tiempos de finalización de tarea esperados y la diferencia entre el plazo de la tarea y los tiempos de finalización. La mínima diferencia (es decir, la relajación de planificación) y el tiempo de finalización final son devueltos al emisor, donde son recibidos en la etapa 401 de recepción (figura 4A).

La figura 5 ilustra la planificación de un procesador que tiene una tarea ejecutándose actualmente, denotada #1, que se inició en $t_{inicio, 1}$. El tiempo actual se denota t_{ahora} . Dos tareas, #2 y #3, están esperando en la cola para ser procesadas. Cada tarea está asociada con un respectivo tiempo de finalización denotado c_1 , c_2 y c_3 . El tiempo de finalización final se denota $t_{ocupado}$ y se calcula como $t_{ocupado} = t_{inicio, 1} + \text{suma}\{c_1, c_2, c_3\}$. La relajación de las tareas se denota mediante sl_1 , sl_2 y sl_3 (del inglés Laxity), respectivamente, y se define como la diferencia entre el tiempo de finalización de la tarea, d_1 , d_2 y d_3 (del inglés Deadline), respectivamente, y el tiempo de finalización de la tarea. La relajación asociada con la planificación como un todo, es decir, la relajación de la planificación, es entonces definida por $sl = \min\{sl_1, sl_2, sl_3\}$ (sl , del inglés Schedule Laxity).

Si la relajación de la planificación se encuentra que es negativa, la función de planificación de tareas local establece que un primer temporizador expire lo más tarde posible cuando es necesario que se inicie la ejecución de la primera tarea de la cola para que la relajación de la planificación sea cero. Este tiempo se denomina t_{ix0} , como se ilustra en la figura 6. La expiración real del temporizador se establece típicamente en algún momento entre t_{ahora} y t_{ix0} . Existen varias maneras de calcular el tiempo de expiración, como se describirá en lo que sigue. Cuando el temporizador expira, la tarea actual vuelve a la función de planificación de tareas global y la primera tarea de la cola en el procesador se activa. La relajación de la planificación y los plazos de finalización son recalculados y, si es necesario, (típicamente una condición rara), se establece un nuevo temporizador.

Aunque es posible establecer que el temporizador expire un tiempo predeterminado antes de t_{ix0} , resulta a menudo más útil calcular un valor que dependa de la situación específica. Por ejemplo, el temporizador puede ser calculado para que expire en una fracción específica del tiempo que queda hasta t_{ix0} , por ejemplo, 3/5 del tiempo que queda, teniendo la fracción específica un valor que tras la sintonización proporciona un rendimiento deseado.

No obstante, en el contexto global, típicamente resulta ventajoso si una tarea actual con una gran relajación es suspendida lo antes posible si una tarea con una relajación considerablemente menor está en la cola. Al mismo

tiempo, no obstante, resulta también ventajoso si una tarea actual con una relajación pequeña realiza el mayor progreso posible antes de ser suspendida. Una manera de conseguir esto es realizar los siguientes cálculos.

5 Si la relajación de la tarea actual es mayor que la relajación de la tarea con la menor relajación de planificación, el temporizador puede ser ajustado para expirar en poco tiempo, por ejemplo 1/5 del tiempo que queda hasta t_{x0} desde t_{ahora} . Si no, el temporizador puede ser ajustado para que expire cerca de t_{x0} , por ejemplo 4/5 del tiempo que queda hasta t_{x0} desde t_{ahora} . Utilizar esta regla mejora la capacidad de planificar cargas de trabajo mayores, en comparación con las situaciones en las que se utilizan reglas de un solo valor.

10 Una alternativa incluso más flexible, pero ligeramente más compleja, es calcular la expiración del temporizador de la siguiente manera. Se calcula la relajación del proceso actual, $l_{x_{actual}}$, seguido del cálculo de la relajación de la tarea con la menor relajación en la planificación, asumiendo que se inicia ahora, $l_{x_{min}}$, y a continuación del cálculo del tiempo de expiración como:

15
$$t_{\text{expiración}} = t_{x0} - (l_{x_{actual}} - l_{x_{min}}) * (t_{x0} - t_{ahora}) / (l_{x_{actual}} + l_{x_{min}})$$

20 La idea de los tiempos de expiración fraccionales duales y de la expiración continua es que resulta globalmente ventajoso si una tarea muy relajada es suspendida lo más pronto posible si existen tareas urgentes. Por otro lado, cuando la relajación de las dos tareas es casi igual, resulta globalmente ventajoso si la tarea actual fuese descartada lo más tarde posible. Los resultados de la simulación muestran que éste es realmente el caso; la expiración del temporizador de acuerdo con las alternativas fraccional dual y continua permiten una utilización casi completa de todos los procesadores aun cumpliendo todos los plazos.

25 Cuando el tiempo de relajación cero se calcula como perteneciente al pasado, la tarea actual puede ser devuelta inmediatamente, si se cumplen las siguientes condiciones: La tarea que indica relajación negativa en la cola y la tarea actual tienen suficiente relajación restante para tolerar un reenvío.

30 Un principio importante es reenviar una tarea sólo si se ha trabajado en ella. Devolver tareas no completamente procesadas puede hacer que la misma tarea rote infinitamente entre una función de planificación de tareas local y la función de planificación de tareas global. En los sistemas de multi-procesadores reales, el tiempo transcurrido entre una decisión de la función de planificación de tareas global y el que la función de planificación de tareas local en un procesador conozca la misma decisión no es cero. Asimismo, el tiempo transcurrido desde una decisión por parte de una función de planificación de tareas local hasta que la función de planificación de tareas global la conoce, tampoco es cero. Tiempos razonables se elevan a cientos de ciclos de reloj de bus. De manera similar, el retardo desde la expiración del temporizador hasta el cambio de tarea real se eleva a cientos de ciclos del procesador.

35 Además, otro retardo es la a veces significativa cantidad de tiempo (típicamente miles de ciclos de reloj) transcurridos cuando se guarda el estado de una tarea de un procesador en un área de memoria común, seguida por la recarga a otro procesador desde la misma área de memoria común.

40 Estos costes de retardos y de migración deben ser tenidos en cuenta en las decisiones de planificación. Afortunadamente, los cálculos de los retardos adicionales pueden ser llevados a cabo por las funciones de planificación de tareas locales. La función de planificación de tareas local calcula y considera un coste adicional cuando la función de planificación de tareas global solicita la devolución de una tarea. Una tarea de la cola es entonces devuelta sólo si se estima que va a cumplir su plazo cuando es procesada por otro procesador (esto añade cuatro retardos de mensaje al tiempo de finalización restante estimado). La función de planificación de tareas local calcula y considera un coste adicional cuando se encuentra que la relajación de planificación es negativa. La tarea actual es a continuación devuelta sólo si se estima que cumple su plazo asumiendo que se necesita una migración completa (esto añade guardar y restaurar tiempos así como cuatro unidades de retardo de mensaje al tiempo de finalización estimado).

45 Ahora con respecto a las llamadas a tarea, como se ha descrito junto con la figura 4B, la función de planificación de tareas global puede emitir una llamada a tarea (etapa 457 de la figura 4B), en el caso de que se descubra que un procesador se ha puesto en reposo. A la recepción de tal solicitud de llamada, la función de planificación de tareas local selecciona una tarea de su cola local para volver a la función de planificación de tareas global.

50 Hay varios criterios de selección alternativos para qué tarea devolver. Puede devolverse la última tarea de la cola. Esta es una manera fácil de asegurar que a la tarea le quedará suficiente relajación para tolerar un reenvío. Puede devolverse también la primera tarea de la cola que tenga suficiente relajación para manejar un reenvío. Una ventaja de tal política es que la mayoría de las tareas urgentes tendrán una oportunidad de un inicio más temprano. Otro criterio de selección es la primera tarea de la cola que tenga un inicio más temprano tras el reenvío, en comparación con permanecer en su cola actual, con respecto a retardos de reenvío.

En la práctica, se ha encontrado que hay poca diferencia entre estos criterios diferentes, simplemente porque es improbable que otro procesador se ponga en reposo cuando hay más de una o quizás dos tareas en la cola.

5 Es posible que una tarea sea estáticamente asignada a un procesador. Las tareas emitidas desde tal tarea forman parte del compromiso del procesador y se utilizan en el cálculo del periodo ocupado y de la relajación. No obstante, puesto que tales tareas son estáticamente asignadas, no pueden ser transmitidas a la función de planificación de tareas global si otro procesador está en reposo.

10 Si la relajación de la planificación se hace negativa con la asignación de una tarea y la tarea actual está asignada estáticamente, el planificador local puede hacer lo siguiente: calcular un tiempo de superposición de tareas como siempre, a continuación calcular el tiempo de finalización para la tarea estáticamente asignada tras su interrupción y reinsertión en la cola, y finalmente la correspondiente relajación de planificación, es decir, un cálculo de una relajación de planificación de asunción dada la asunción de que la tarea que se está ejecutando actualmente está replanificada. Si la relajación resultante es positiva, el temporizador es ajustado (como se ha explicado anteriormente) y, si es negativa, no se ajusta ningún temporizador. A la expiración del temporizador la tarea estáticamente asignada es reinsertada en la cola local.

20 Debe observarse que, en presencia de una tarea estáticamente asignada, la función de planificación de tareas local no debe devolver una tarea dinámica, que se acaba de asignar al emisor global, puesto que el emisor probablemente asignaría la tarea de nuevo al mismo procesador. Aunque puede considerarse que otra tarea distinta de la que se acaba de asignar pudiese ser devuelta, esto probablemente tendría una consecuencia similar.

25 Una característica inherente de los algoritmos de planificación que están basados en alguna forma de orden de plazo es que proporcionan la prioridad más alta a las tareas que están cerca o no han cumplido su plazo. El resultado de esto es que es imposible predecir o restringir los incumplimientos de plazo para ciertas tareas. Esto es altamente indeseable para sistemas críticos en seguridad y de alta fiabilidad.

30 Una manera de evitar esto es asignar una prioridad de tareas, que exprese cómo es de crítica la tarea para el sistema en el cual va a ser ejecutada, o qué importancia tiene el cumplimiento del plazo de la tarea. Los algoritmos explicados anteriormente pueden ser fácilmente extendidos para manejar esto de la siguiente manera.

35 Para cada función de planificación de tareas local, hay una cola para cada nivel de prioridad de tarea. El número de niveles de prioridad debe ser al menos dos. Las tareas que entran en una cola de mayor prioridad que la prioridad de la tarea actual provocan una interrupción si es necesario de la misma manera que se ha explicado anteriormente en conexión con la figura 6. El tiempo de finalización de planificación y la relajación se calcula para cada nivel de prioridad y la función de planificación de tareas local guarda el tiempo de finalización de planificación y tablas de relajación para todos los niveles de prioridad. Las tareas de alta prioridad son enviadas antes que las tareas de menor prioridad.

40 Tal algoritmo puede ser también complementado con una planificación ordenada en el nivel de prioridad más bajo para tareas de segundo plano que comparten tiempo.

45 Volviendo ahora a las figuras 7a-c y 8a-e, se describirá la planificación que implica un cálculo del temporizador fraccional dual por medio de diagramas de flujo que ilustran etapas tomadas en una función de planificación de tareas global y en funciones de planificación de tareas locales, tales como la función de planificación de tareas global y las funciones de planificación de tareas locales descritas anteriormente en conexión con las figuras 1 a 6.

50 La figura 7a muestra las etapas llevadas a cabo durante el envío de tareas llevado a cabo en la función de planificación de tareas global. En una etapa de comprobación 701 se realiza una comprobación acerca de si hay algunas tareas en la cola y si hay algún procesador en reposo o válido disponible. Un procesador es considerado como válido si el procesador ha reconocido un envío de proceso o un evento similar. Si ese es el caso, se lleva a cabo una etapa de selección 705. En la etapa de selección 705, la primera tarea en la cola de envío global es seleccionada para su envío. La función de planificación de tareas global tiene una cola que contiene tareas que no han sido enviadas todavía (típicamente porque la mayoría de los procesadores no ha reconocido todavía el último envío cuando ha llegado una tarea). Si no se ha identificado ningún procesador en reposo y válido en la etapa de comprobación 701, el flujo del proceso vuelve a alguna aplicación de llamada, como se indica mediante una etapa de devolución 703.

60 En una etapa de comprobación 707 se comprueba si hay al menos un procesador en reposo y, si es así, la tarea es enviada, en una etapa de envío 709, al primer procesador en reposo. Una etapa de actualizar el estatus del procesador es llevada a cabo a continuación en una etapa de actualización 711. Durante esta etapa el estatus del procesador es invalidado. Para un procesador en reposo, esto conlleva eliminarlo de la lista de procesadores en reposo. La etapa de actualización 711 es seguida por una eliminación 725 de la tarea de la cola.

5 Si no se ha encontrado ningún procesador en reposo en la etapa 707, la relajación de tarea se calcula en una etapa de cálculo 713 y se obtiene el tiempo de finalización válido más temprano, en una etapa de obtención 715, de la lista de pares {tiempo de finalización válido, referencia de procesador}. La lista puede ser guardada clasificada según tiempos de finalización ascendentes, en cuyo caso ésta es una lectura del primer elemento de esa lista. La relajación se compara con el tiempo de finalización válido más temprano en una etapa de comprobación 717 y, si la relajación es mayor, entonces la tarea es enviada en una etapa de envío 719 al procesador de finalización más temprana. Si la relajación no es mayor que el tiempo de finalización válido más temprano, la tarea es enviada en una etapa de envío 721. A continuación se lleva a cabo una etapa de actualización 723 del estatus del procesador. De manera similar a la etapa 711, el estatus del procesador es invalidado de manera que los pares {tiempo de finalización, ref de procesador} y {relajación de planificación, ref de procesador} son eliminados de las listas de tiempos de finalización y de relajaciones respectivamente. A continuación la tarea es eliminada de la cola en la etapa de eliminación 725.

10 La figura 7b muestra las etapas llevadas a cabo cuando una tarea llega a la función de planificación de tareas global. En una etapa de puesta en cola 731, la tarea es puesta en una cola. A continuación, en una etapa de comprobación 733 se realiza una comprobación de si hay algún procesador en reposo y válido disponible. Si ese es el caso, las tareas de la cola son enviadas 735 como se describe mediante el diagrama de flujo de la figura 7a.

15 La figura 7c muestra las etapas llevadas a cabo cuando se recibe un reporte de estatus desde una función de planificación de tareas local. En una etapa de comprobación 741 se comprueba si una tarea es devuelta y, si es así, se realiza una comprobación en una etapa de comprobación 745 acerca de si la tarea devuelta ha finalizado o no. Si la tarea devuelta ha finalizado, la tarea es retirada en una etapa de retirada 749 y si la tarea no está finalizada es reinsertada en la cola en una etapa de nueva puesta en cola 747. Si se ha encontrado que no se ha devuelto ninguna tarea en la etapa de comprobación 741, el estatus del procesador es actualizado de tal manera que el procesador es marcado como válido (si no ha sido ya marcado de tal manera) en una etapa de actualización de estatus 743.

20 A continuación se realiza una comprobación en una etapa de comprobación 751 acerca de si hay alguna tarea en la cola y de si hay algún procesador en reposo o el número de procesadores válidos es mayor que un número mínimo. Si es así, las tareas son enviadas 753 como se describe mediante el diagrama de flujo de la figura 7a. A continuación se realiza una comprobación en una etapa de comprobación 755 acerca de si hay algún procesador en reposo y si el número de tareas enviadas es mayor que el número de procesadores. Si es así, se emite una solicitud de llamada, en una etapa de llamada 757, al procesador con el mayor número de tareas.

25 La figura 8a muestra las etapas llevadas a cabo en una función de planificación de tareas local cuando se recibe una solicitud de llamada desde la función de planificación de tareas global. Se realiza una comprobación 801 y sólo si hay más de una tarea, el procedimiento continúa. Se realiza una selección en una etapa de selección 803 de la tarea que está asociada con la mayor relajación de la cola, por ejemplo, la última tarea de la cola. En una etapa de cálculo 805, se calcula el coste de migración, es decir, que implica retardos, etc. como se ha descrito anteriormente. Sólo si la relajación es mayor que el coste de migración, como se ha comprobado en una etapa de comprobación 807, el procedimiento continúa. La tarea seleccionada es eliminada de la cola en una etapa de eliminación 809 tras la cual la relajación y el tiempo de finalización de planificación se calculan en una etapa de cálculo 811. Si la relajación de planificación es menor de cero, como se ha comprobado en la etapa de comprobación 813, se inicia un temporizador de replanificación en una etapa de activar temporizador 815, como se describirá con más detalle en lo que sigue. A continuación la tarea seleccionada es devuelta a la función de planificación de tareas global junto con el estatus del procesador en una etapa de devolver tarea 817.

30 La figura 8b muestra las etapas llevadas a cabo en una función de planificación de tareas local cuando se recibe una tarea de la función de planificación de tareas global. La tarea es puesta en cola en una etapa de poner en cola 821. Si el procesador está activo, como se ha comprobado en una etapa de comprobación 823, la primera tarea de la cola es activada en una etapa de activación 825. Si el procesador está activo, como se ha comprobado en una etapa de comprobación 823, la primera tarea de la cola es activada en una etapa de activación 825. Si el procesador no está activo, el tiempo de finalización de planificación y la relajación se calculan en una etapa de cálculo 827. Si la relajación de planificación es menor de cero, como se ha comprobado en una etapa de comprobación 829, un temporizador de replanificación es iniciado en una etapa de activar temporizador 831, como se describirá con más detalle a continuación. A continuación el estatus del procesador es enviado a la función de planificación de tareas global en una etapa de enviar estatus 833.

35 La figura 8c muestra etapas llevadas a cabo en una función de planificación de tareas local cuando expira un temporizador de replanificación. En una etapa de desactivación 841, la tarea actual es desactivada y eliminada de la cola. La primera tarea de la cola es a continuación activada en una etapa de activación 843. A continuación el tiempo de finalización de planificación y la relajación se calculan en una etapa de cálculo 845. Si la relajación de planificación es menor de cero, como se comprueba en una etapa de comprobación 847, se inicia un temporizador de replanificación en una etapa de activar temporizador 849, como se describirá con más detalle a continuación.

Entonces la primera tarea actual es devuelta a la función de planificación de tareas global junto con el estatus del procesador en una etapa de devolver tarea 851.

- 5 La figura 8d muestra etapas llevadas a cabo en una función de planificación de tareas local cuando una tarea finaliza. La tarea finalizada es eliminada de la cola en una etapa de eliminación 861. A continuación se comprueba en una etapa de comprobación 863 si hay alguna tarea en la cola y, si es así, se activa la primera tarea de la cola en una etapa de activación 865. Si no existe ninguna tarea en la cola, como se ha comprobado en la etapa de comprobación 863, se calculan el tiempo de finalización de planificación y la relajación en una etapa de cálculo 867. Esto es, el tiempo de finalización se pone en el tiempo actual, y la relajación se pone en una representación de infinito. Si la relajación de planificación es menor de cero, como se ha comprobado en una etapa de comprobación 10 869, se inicia un temporizador de replanificación en una etapa de activar temporizador 871, como se describirá con más detalle en lo que sigue. A continuación la tarea finalizada es devuelta a la función de planificación de tareas global junto con el estatus del procesador en una etapa de devolver tarea 873.
- 15 Volviendo ahora a la figura 8e, se describirá un procedimiento de temporizador que tiene lugar en una función de planificación de tareas local. El procedimiento del temporizador de la figura 8e se inicia como se ha descrito anteriormente en conexión con cualquiera de las figuras 8a-d. Si ya se ha puesto, el temporizador es cancelado en una etapa de cancelar temporizador 881. Entonces, sólo si hay más de una tarea en la cola, como se ha comprobado en una etapa de comprobación 883, el procedimiento del temporizador continúa. El coste de migración relativo a la tarea que se está ejecutando actualmente es calculado en una etapa de cálculo 885 y es comparado, en una etapa de comprobación 887, con la relajación de la tarea actual. El proceso continúa sólo si la relajación de la tarea actual es mayor que el coste de migración calculado. A continuación el procedimiento continúa con el cálculo, en una etapa de cálculo 889, de un momento del tiempo de expiración añadiendo la relajación de planificación al tiempo de finalización de la tarea que se está ejecutando actualmente. Este momento del tiempo de expiración es a continuación comparado, en una etapa de comprobación 891, con el tiempo actual y, si no es más tarde que el tiempo actual, se ajusta un momento del tiempo de expiración para la tarea que se está ejecutando actualmente, en una etapa de ajuste 893, en el tiempo actual, tras lo cual el temporizador es ajustado a este momento del tiempo de expiración en una etapa de ajuste 987.
- 30 Por otro lado, si se hubiese encontrado en la etapa de comprobación 891, que el momento del tiempo de expiración calculado es posterior al tiempo actual, se realiza una determinación, en una etapa de cálculo 892, acerca de la relajación de la tarea actual y de una relajación mínima. Siendo la relajación mínima la relajación de la tarea que tiene la menor relajación de planificación. Estos valores son entonces comparados, en una etapa de comprobación 894, y si la relajación de la tarea actual no es mayor que la relajación de la tarea que tiene la menor relajación de planificación, entonces se establece un momento del tiempo de expiración para la tarea que se está ejecutando actualmente, en una etapa de ajuste 896, al tiempo actual más una primera fracción predeterminada de la diferencia entre el momento del tiempo de expiración calculado y el tiempo actual, tras lo cual el temporizador es ajustado a este momento del tiempo de expiración en la etapa de ajuste 897. Si, en la etapa de comprobación 894, se encuentra que la relajación de la tarea actual es mayor que la relajación de la tarea que tiene la menor relajación de planificación, entonces se establece un momento del tiempo de expiración para la tarea que se está ejecutando actualmente, en una etapa de ajuste 895, al tiempo actual más una segunda fracción predeterminada de la diferencia entre el momento del tiempo de expiración calculado y el tiempo actual, tras el cual el temporizador es ajustado a este momento del tiempo de expiración en la etapa de ajuste 897. Los valores de las fracciones primera y segunda predeterminadas pueden ser seleccionados de tal manera que el fraccional en la etapa 895 esté entre 0,05 y 0,2, mientras que la fracción en la etapa 896 está entre 0,6 y 0,9. En simulaciones de la realización se han utilizado los valores $2/16$ (0,125) y $11/16$ (0,6875), respectivamente, con excelentes resultados. La evidencia empírica sugiere que las fracciones en los intervalos dados funcionan muy bien para muchos conjuntos de tareas.
- 45

REIVINDICACIONES

- 5 1. Un método de planificar tareas de procesamiento en un aparato de cálculo de multi-procesadores (100, 200), estando las tareas asociadas con al menos un plazo respectivo, implicando el método la comunicación entre una función de planificación de tareas global (102, 202) y una pluralidad de funciones de planificación de tareas locales (112, 212), estando cada función de planificación de tareas local asociada con un procesador (108, 208) respectivo, que comprende:
- 10 - guardar, en la función de planificación de tareas global, información del estatus actualizada relativa a :
- cualquier procesador en reposo,
 - momentos del tiempo en los que se espera que cualquier procesador esté en reposo,
 - valores de relajación asociados con cualquier procesador,
- 15 comprendiendo también el método:
- planificar, en la función de planificación de tareas global, las tareas de:
- 20 - enviar tareas a las funciones de planificación de tareas locales, estando el envío condicionado por la información de estatus,
- llamar a tareas desde las funciones de planificación de tareas locales en algún procesador que no esté en reposo y enviar alguna tarea llamada a una función de planificación de tareas local en un procesador que no esté en reposo,
- 25 - recibir, desde las funciones de planificación de tareas locales, tareas interrumpidas y enviar cualquier tarea interrumpida recibida a una función de planificación de tareas local, estando el envío condicionado a la información de estatus,
- donde el hecho de guardar la información de estatus actualizada comprende:
- 30 - recibir, desde cada función de planificación de tareas local, un valor de relajación de planificación calculado respectivo que es la diferencia mínima entre los plazos de finalización para tareas y los tiempos de finalización de tarea esperados con tareas actualmente enviadas a cada procesador respectivo, y
- donde la planificación de una tarea comprende:
- 35 - comprobar si existe un procesador en reposo y, si es así, enviar la tarea a la función de planificación de tareas local del procesador en reposo,
- 40 si no:
- calcular un tiempo de finalización esperado de la tarea, siendo el cálculo relativo a una situación que implica a la existencia de un procesador en reposo,
 - calcular un valor de relajación de tarea como una diferencia entre el plazo para la tarea y el tiempo de finalización esperado calculado para la tarea,
- 45 - comprobar si la relajación de la tarea calculada es mayor que una diferencia entre el momento del tiempo más temprano en el que algún procesador se espera que esté en reposo y el tiempo actual y, si es así, enviar la tarea a la función de planificación de tareas local de ese procesador,
- 50 si no:
- enviar la tarea a la función de planificación de tareas local del procesador que tenga el mayor valor de relajación de planificación.
- 55 2. Un método de planificar tareas de procesamiento en un aparato de cálculo de multi-procesadores (100, 200), estando las tareas asociadas con al menos un plazo respectivo, implicando el método la comunicación entre una función de planificación de tareas global (102, 202) y una pluralidad de funciones de planificación de tareas locales (112, 212), estando cada función de planificación de tareas local asociada con un procesador (108, 208) respectivo, que comprende:
- 60 - guardar, en la función de planificación de tareas global, información de estatus actualizada relativa a :
- cualquier procesador en reposo,
 - momentos del tiempo en los que algún procesador se espera que esté en reposo,
 - valores de relajación asociados con algún procesador,

comprendiendo también el método:

- planificar, en la función de planificación de tareas global, tareas mediante:

- 5
- enviar tareas a las funciones de planificación de tareas locales, estando el envío condicionado por la información de estatus,
 - llamar a tareas desde las funciones de planificación de tareas locales en cualquier procesador que no esté en reposo y enviar alguna tarea llamada a una función de planificación de tareas local en un procesador en reposo,
- 10
- recibir, desde las funciones de planificación de tareas locales, tareas interrumpidas y enviar cualquier tarea interrumpida recibida a una función de planificación de tareas local, estando el envío condicionado por la información de estatus,

donde el hecho de guardar información de estatus actualizada comprende:

- 15
- recibir, desde cada función de planificación de tareas local, un valor de relajación de planificación calculado respectivo que es una diferencia mínima entre los plazos de finalización para las tareas y los tiempos de finalización de tarea esperados asociados con las tareas enviadas actualmente a cada procesador respectivo,
- 20
- y

donde la planificación de una tarea comprende:

- 25
- comprobar si existe un procesador en reposo y, si es así, enviar la tarea a la función de planificación de tareas local del procesador en reposo,

si no:

- 30
- comprobar si un plazo de inicio para la tarea es posterior al momento del tiempo más temprano en el que se espera que cualquier procesador esté en reposo y, si es así, enviar la tarea a la función de planificación de tareas local de ese procesador,

si no:

- 35
- enviar la tarea a la función de planificación de tareas local del procesador que tenga el valor de relajación de planificación más grande.

3. El método de cualquiera de las reivindicaciones 1 a 2, que comprende, en la función de planificación de tareas global:

- 40
- determinar a partir de la información de estatus actualizada si al menos un procesador está en reposo y, si es así:
 - determinar los datos de decisión de llamada que comprenden información relativa al número total de tareas que se están enviando actualmente a las funciones de planificación de tareas locales, el número de procesadores que no están en reposo e información relativa a cuál de los procesadores es el más ocupado,
- 45
- decidir, utilizando los datos de decisión de llamada, si llamar a una tarea desde el procesador más ocupado y, si es así:

- 50
- recibir una tarea desde el procesador más ocupado y,
 - enviar la tarea recibida a la función de planificación de tareas local de un procesador en reposo.

4. El método de la reivindicación 3, que comprende, en una función de planificación de tareas local:

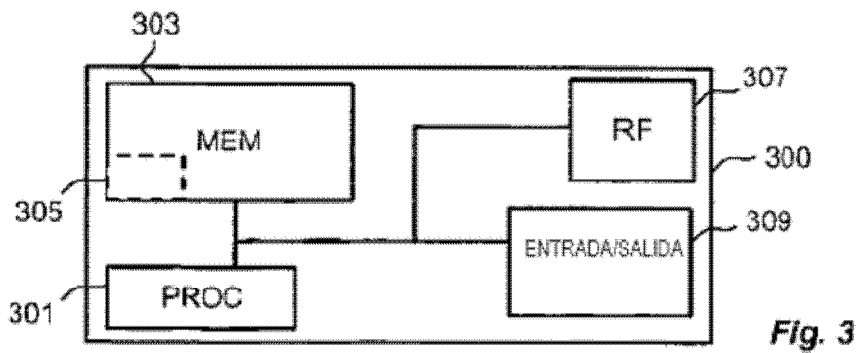
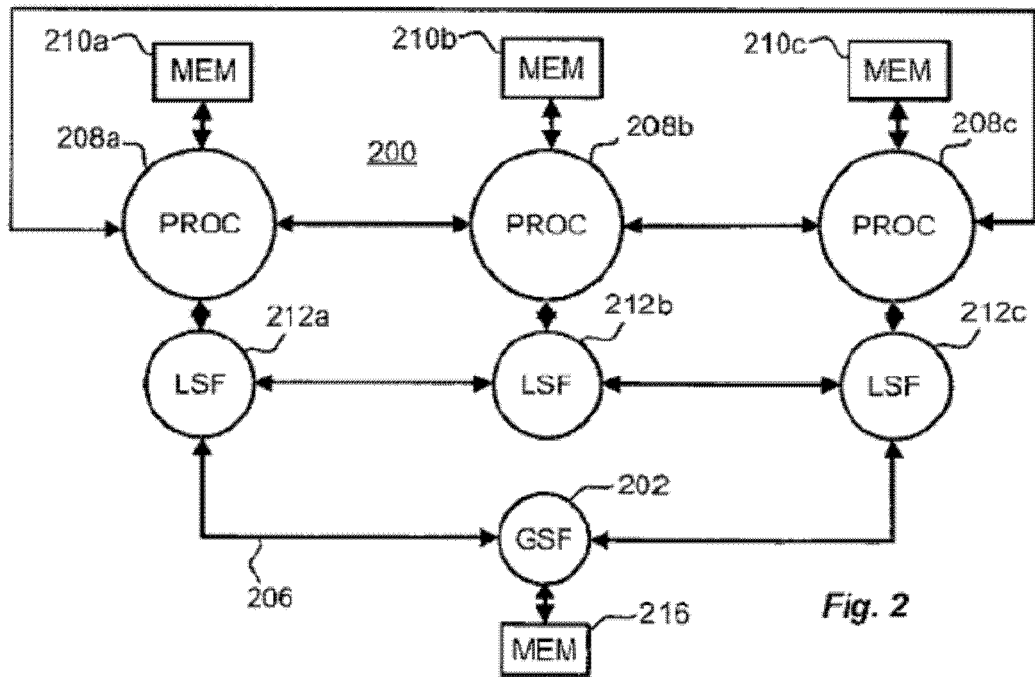
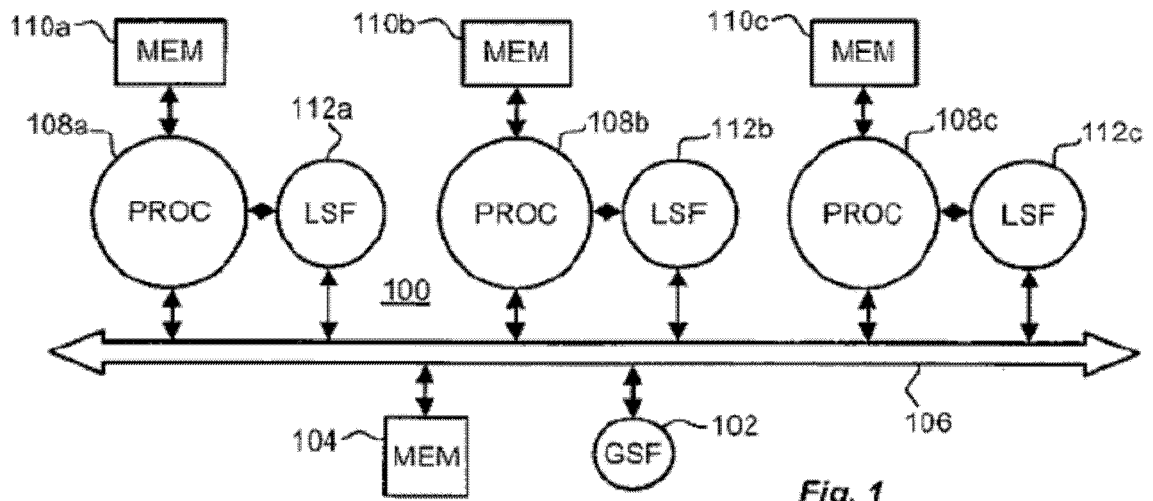
- 55
- recibir una solicitud de llamada de tarea desde la función de planificación de tareas global,
 - estimar si las tareas de la cola pueden cumplir sus respectivos plazos de finalización si son procesados por algún procesador y, dependiendo de la estimación:
 - devolver una tarea a la función de planificación de tareas global.

5. El método de la reivindicación 4, que comprende, en una función de planificación de tareas local:

- 60
- determinar devolver los criterios de selección para decidir qué tarea devolver a la función de planificación de tareas global, comprendiendo la determinación de devolver los criterios de selección cualquiera de:
 - una tarea última en una cola de tareas,

- 5
- calcular el reenvío de valores de retardo para tareas de la cola y comparar los valores de retardo de reenvío con valores de relajación de las tareas de la cola,
 - calcular valores de retardo de reenvío para tareas de la cola y determinar un tiempo de inicio respectivo, teniendo en cuenta los valores de retardo de reenvío, en cualquier procesador y comparar estos tiempos de inicio, y a continuación determinar los criterios de selección de devolución:
- 10
- devolver una tarea a la función de planificación de tareas global de acuerdo con los criterios de selección de devolución.
- 10
6. El método de cualquiera de las reivindicaciones 1 a 5, que comprende, en una función de planificación de tareas local:
- 15
- comprobar si la relajación de planificación es menor de cero, y si es así:
 - calcular un momento del tiempo de expiración que sea el momento del tiempo en el cual una tarea que se está ejecutando actualmente debe detenerse con el fin de que la relajación de planificación sea cero,
 - interrumpir la tarea que se está ejecutando actualmente, teniendo lugar la interrupción en un intervalo de tiempo que es distinto de cero antes del momento del tiempo de expiración,
 - devolver la tarea interrumpida a la función de planificación de tareas global,
 - activar una tarea de la cola, y
 - recalcular la relajación de planificación.
- 20
7. El método de la reivindicación 6, en el que el intervalo de tiempo distinto de cero se determina:
- 25
- calculando la relajación de la tarea que se está ejecutando actualmente,
 - calculando una relajación de asunción para cada tarea de la cola, como la diferencia entre un tiempo de finalización esperado y un plazo de finalización con respecto a un momento del tiempo de inicio asumido común,
 - identificando la menor relajación de asunción entre las tareas de la cola,
 - ajustando el intervalo de tiempo distinto de cero, en caso de que la relajación de la tarea que se está ejecutando actualmente sea mayor que la menor relajación de asunción identificada, para un primer valor del intervalo del temporizador, y si no:
 - ajustando el intervalo de tiempo distinto de cero a un segundo valor del intervalo del temporizador, siendo el segundo valor del intervalo del temporizador mayor que el primer valor del intervalo del temporizador.
- 30
- 35
8. El método de la reivindicación 6, en el que el intervalo de tiempo distinto de cero se determina:
- 40
- calculando la relajación de la tarea que se está ejecutando actualmente,
 - calculando una relajación de asunción para cada tarea de la cola, como la diferencia entre un tiempo de finalización esperado y un plazo de finalización con respecto a un momento del tiempo de inicio asumido común,
 - identificando la menor relajación de asunción entre las tareas de la cola,
 - ajustando el intervalo de tiempo distinto de cero a un valor que es proporcional a la diferencia entre la relajación de la tarea que se está ejecutando actualmente y la menor relajación de asunción e inversamente proporcional a la suma de la relajación de la tarea que se está ejecutando actualmente y la menor relajación de asunción.
- 45
9. El método de cualquiera de las reivindicaciones 6 a 8, que comprende, en una función de planificación de tareas local:
- 50
- recibir una solicitud de llamada de tarea desde la función de planificación de tareas global,
 - estimar si las tareas de la cola pueden cumplir sus respectivos plazos de finalización teniendo en cuenta cualquier retardo provocado por salvar y restaurar los datos asociados con la replanificación entre procesadores y, dependiendo de la estimación:
 - devolver una tarea a la función de planificación de tareas global.
- 55
10. El método de cualquiera de las reivindicaciones 1 a 5, que comprende, en una función de planificación de tareas local:
- 60
- comprobar si la relajación de planificación es menor de cero y, si es así:
 - calcular un momento del tiempo de expiración, siendo el momento del tiempo en el cual una tarea que se está ejecutando actualmente debe detener su ejecución con el fin de que la relajación de planificación sea cero,
 - calcular una relajación de planificación de asunción dada la asunción de que la tarea que se está ejecutando actualmente es replanificada,

- interrumpir, si la relajación de planificación de asunción calculada es positiva, la tarea que se está ejecutando actualmente en un intervalo de tiempo distinto de cero antes del momento del tiempo de expiración, y si no interrumpir la tarea que se está ejecutando actualmente inmediatamente,
 - activar una tarea de la cola, y
 - recalcular la relajación de planificación.
- 5
11. El método de cualquiera de las reivindicaciones 1 a 10, que comprende, en una función de planificación de tareas local:
- asignar un nivel de prioridad a cada tarea,
 - guardar una pluralidad de colas de tareas, conteniendo cada cola tareas que tienen un único nivel de prioridad, y
 - guardar, en la función de planificación de tareas global, información de estatus actualizada relativa a la pluralidad de colas.
- 10
- 15
12. El método de cualquiera de las reivindicaciones 1 a 11, en el que la función de planificación de tareas global está asociada con un procesador específico.
- 20
13. El método de cualquiera de las reivindicaciones 1 a 11, en el que la función de planificación de tareas global está asociada con una pluralidad de procesadores.
- 25
14. El método de cualquiera de las reivindicaciones 1 a 13, en el que el hecho de guardar la información de estatus actualizada comprende:
- guardar una lista de procesadores clasificada en orden de momentos del tiempo ascendentes en los que se espera que los procesadores estén en reposo, y
 - guardar una lista de procesadores clasificada en orden de relajación descendente.
- 30
15. El método de la reivindicación 14, en el que el hecho de guardar la información de estatus actualizada comprende también:
- guardar una lista de primero en entrar primero en salir de procesadores en reposo.
- 35
16. Un programa de ordenador que comprende instrucciones de software que, cuando son ejecutadas en un dispositivo de procesamiento de multi-procesadores, lleva a cabo el método de cualquiera de las reivindicaciones 1 a 15.
- 40
17. Un dispositivo de procesamiento de multi-procesadores (100, 200, 300) que comprende un medio de procesamiento y de memoria (108, 208, 301, 303, 305) que están configurados de tal manera que es capaz de llevar a cabo el método de cualquiera de las reivindicaciones 1 a 15.
18. Un dispositivo de comunicación de telefonía móvil (300) que comprende el dispositivo de procesamiento de multi-procesadores de la reivindicación 17.



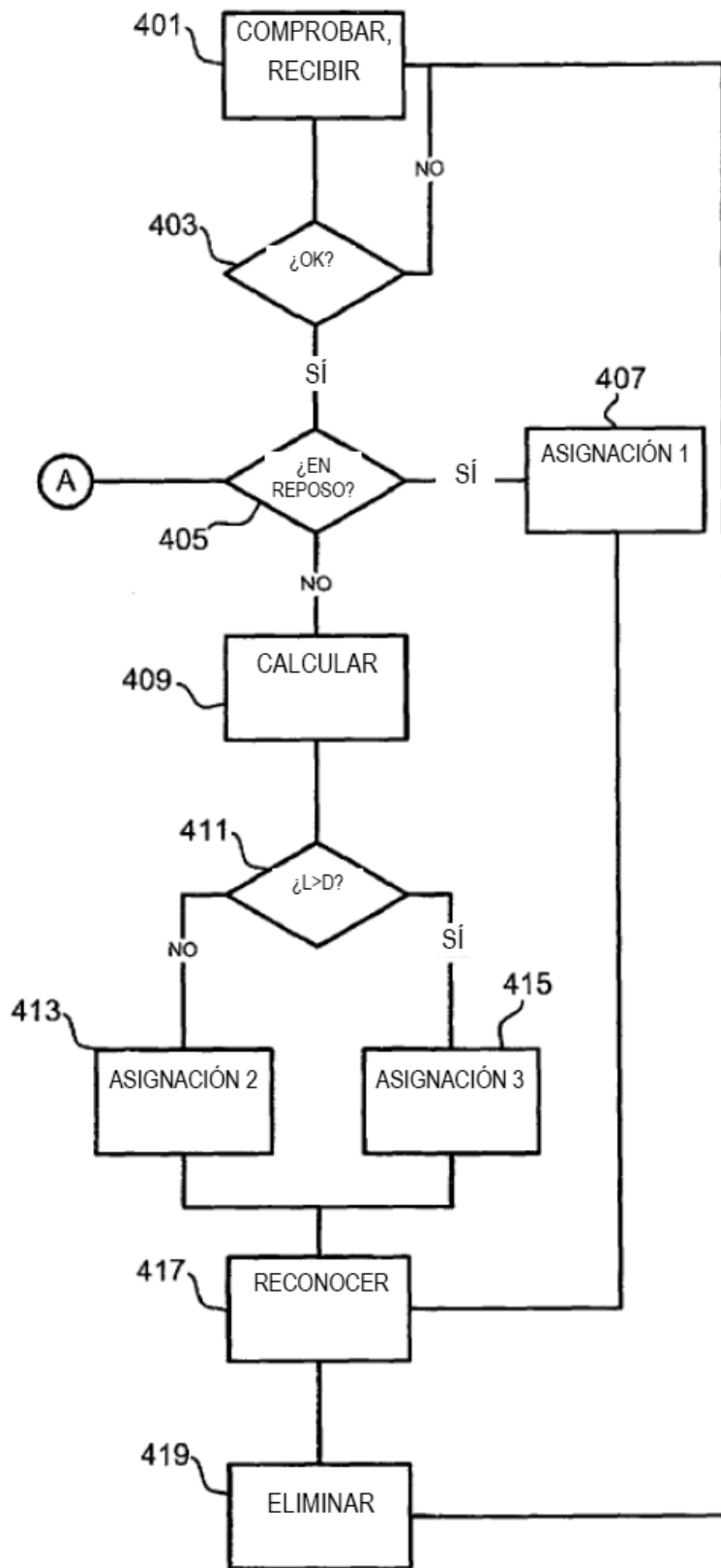


Fig. 4A

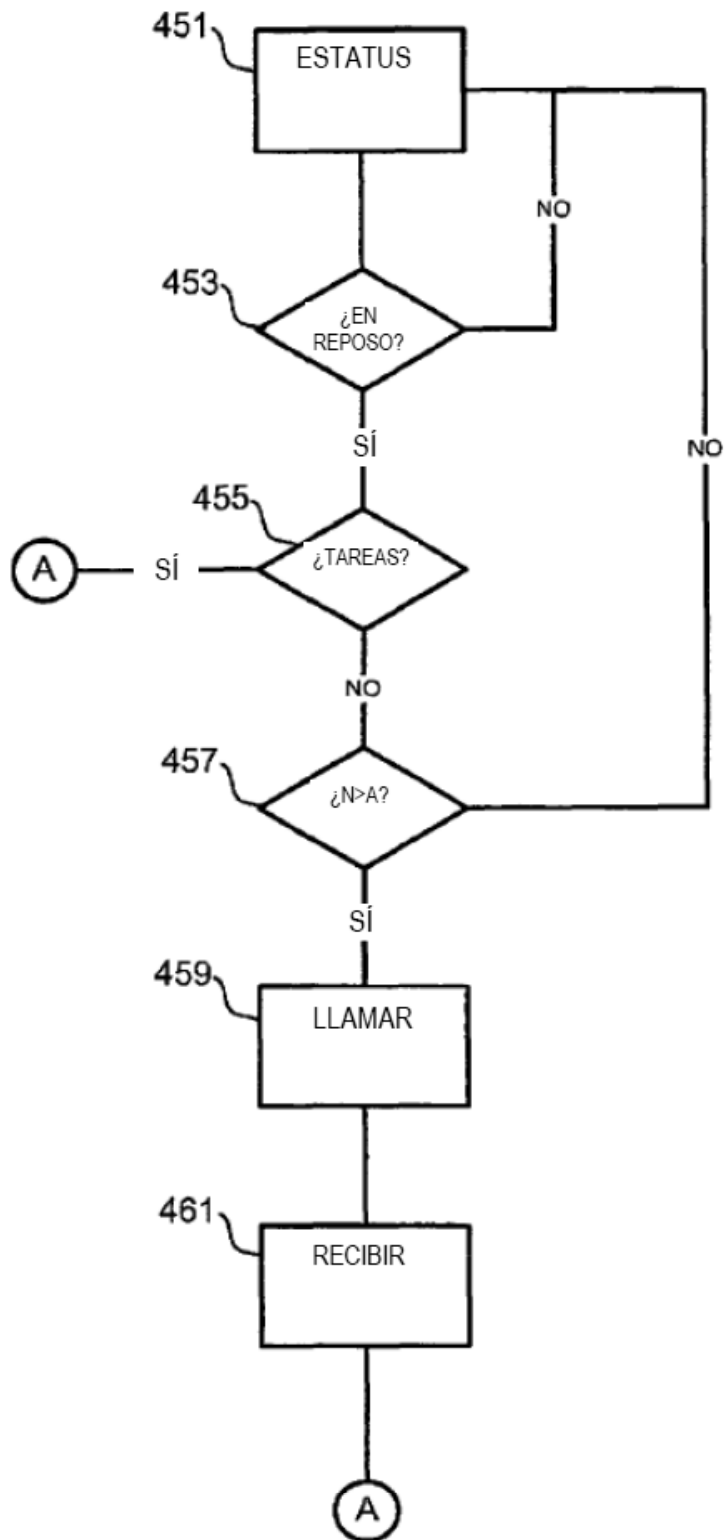


Fig. 4B

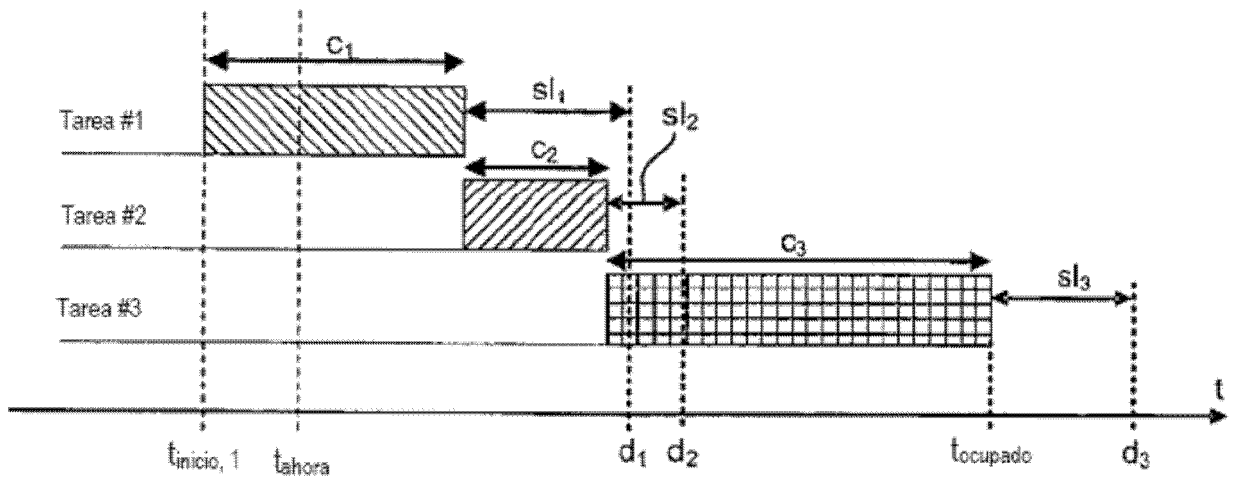


Fig. 5

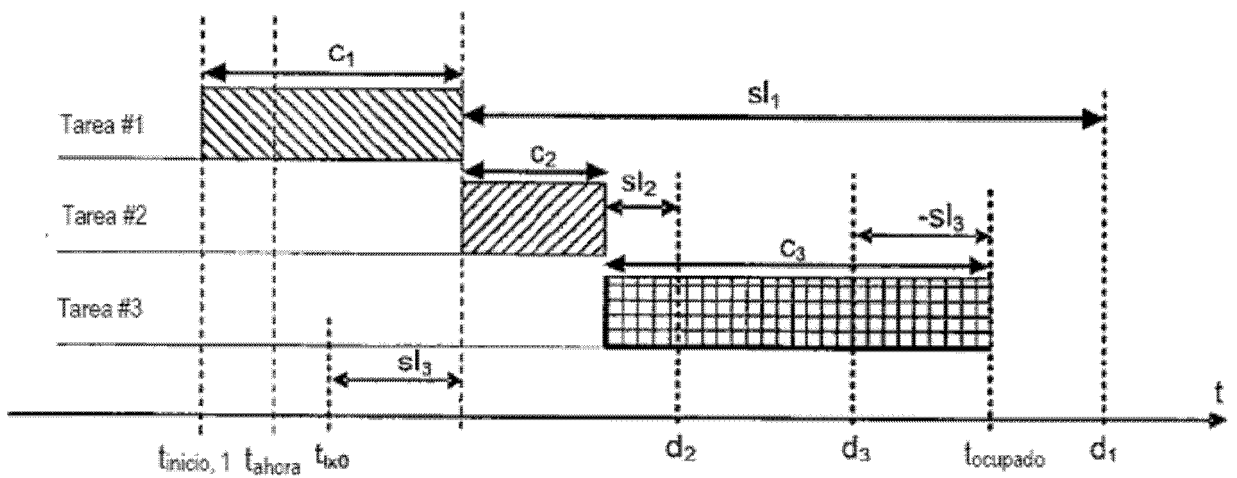
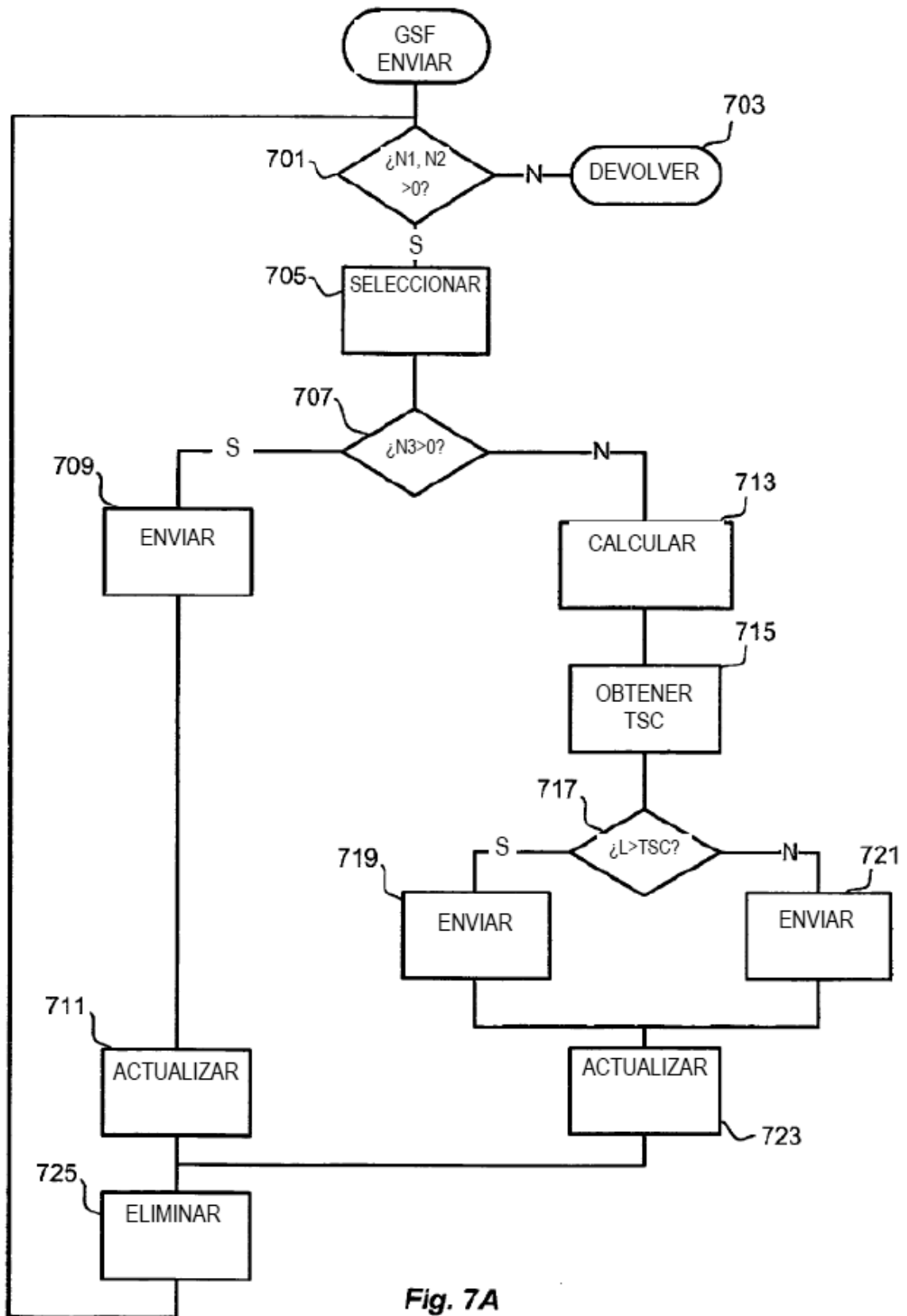


Fig. 6



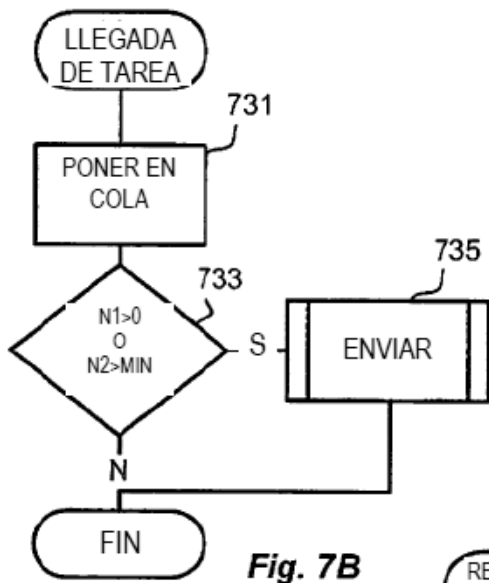


Fig. 7B

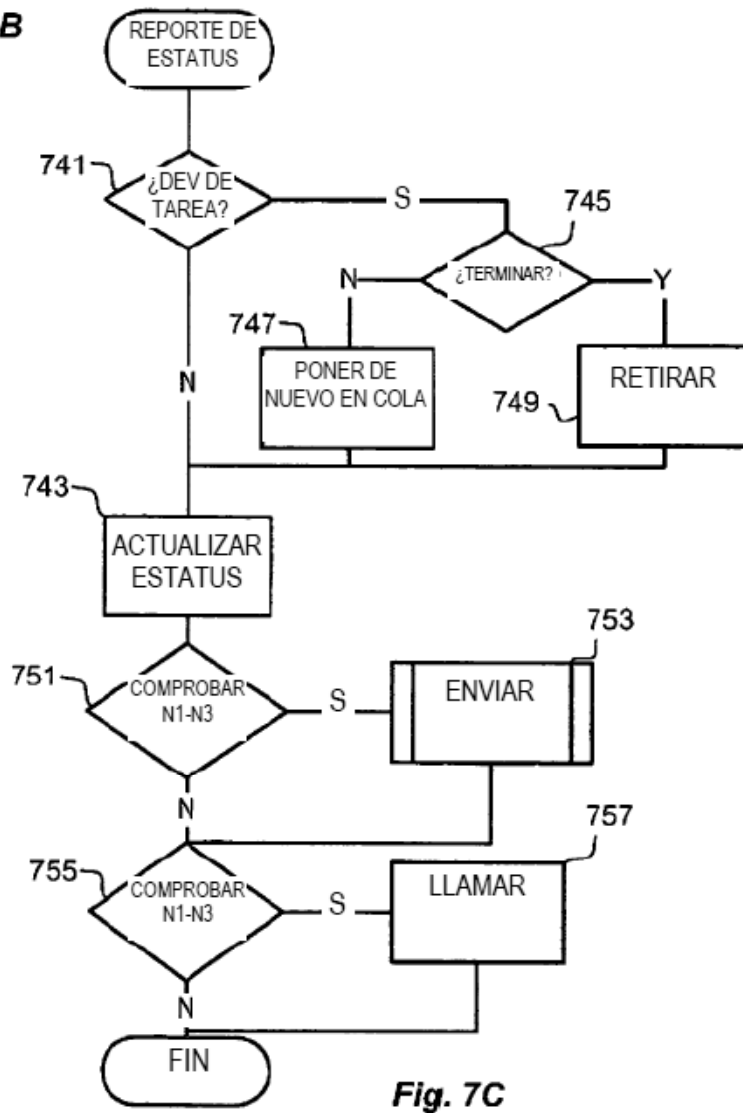


Fig. 7C

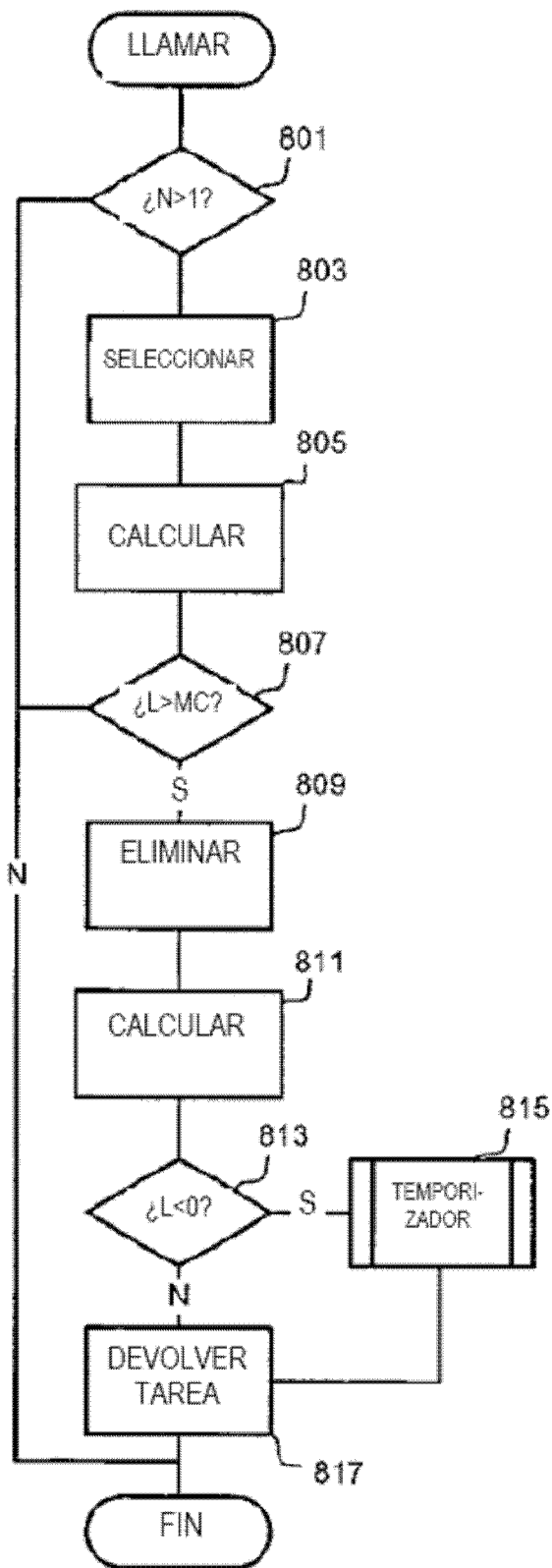


Fig. 8A

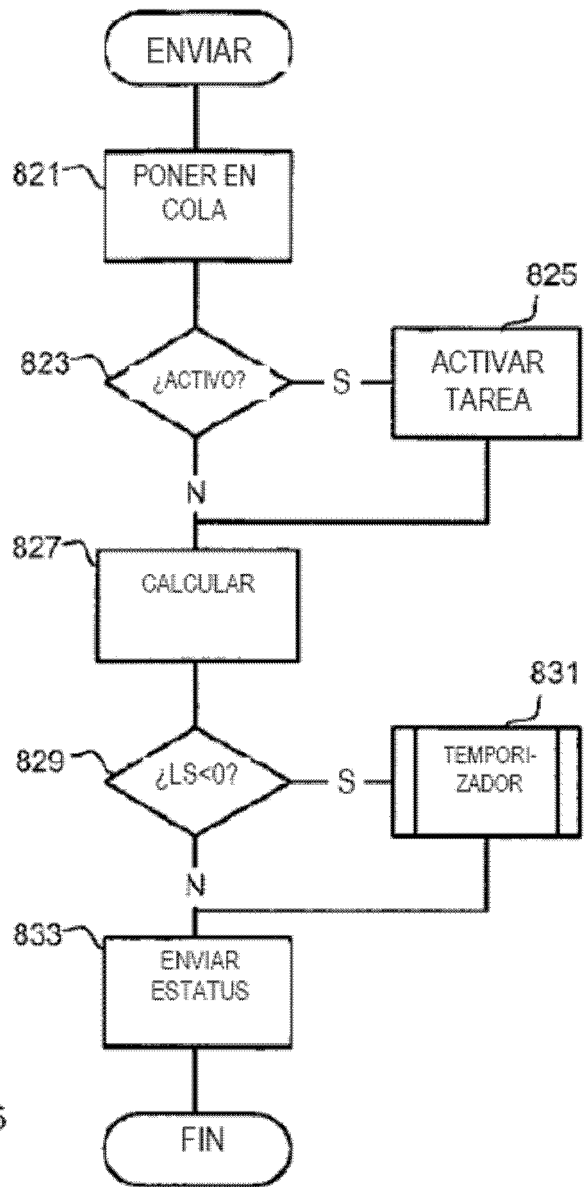


Fig. 8B

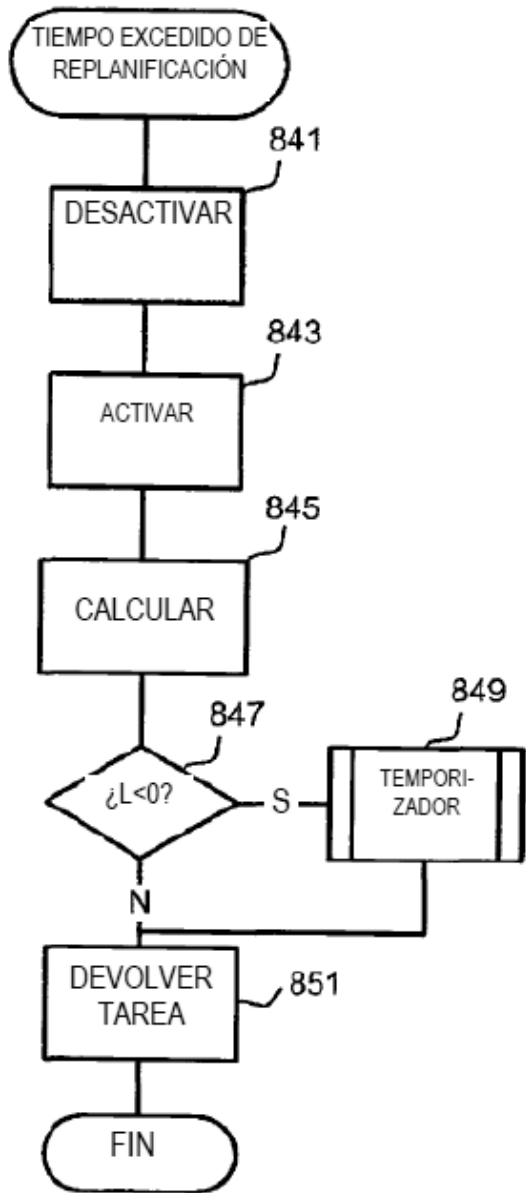


Fig. 8C

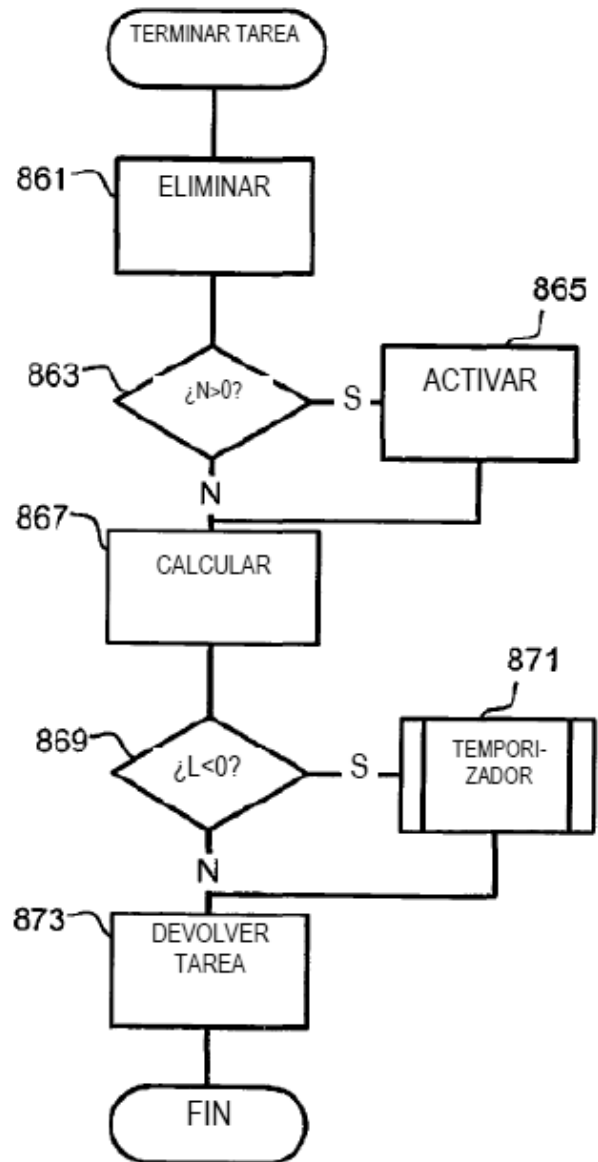


Fig. 8D

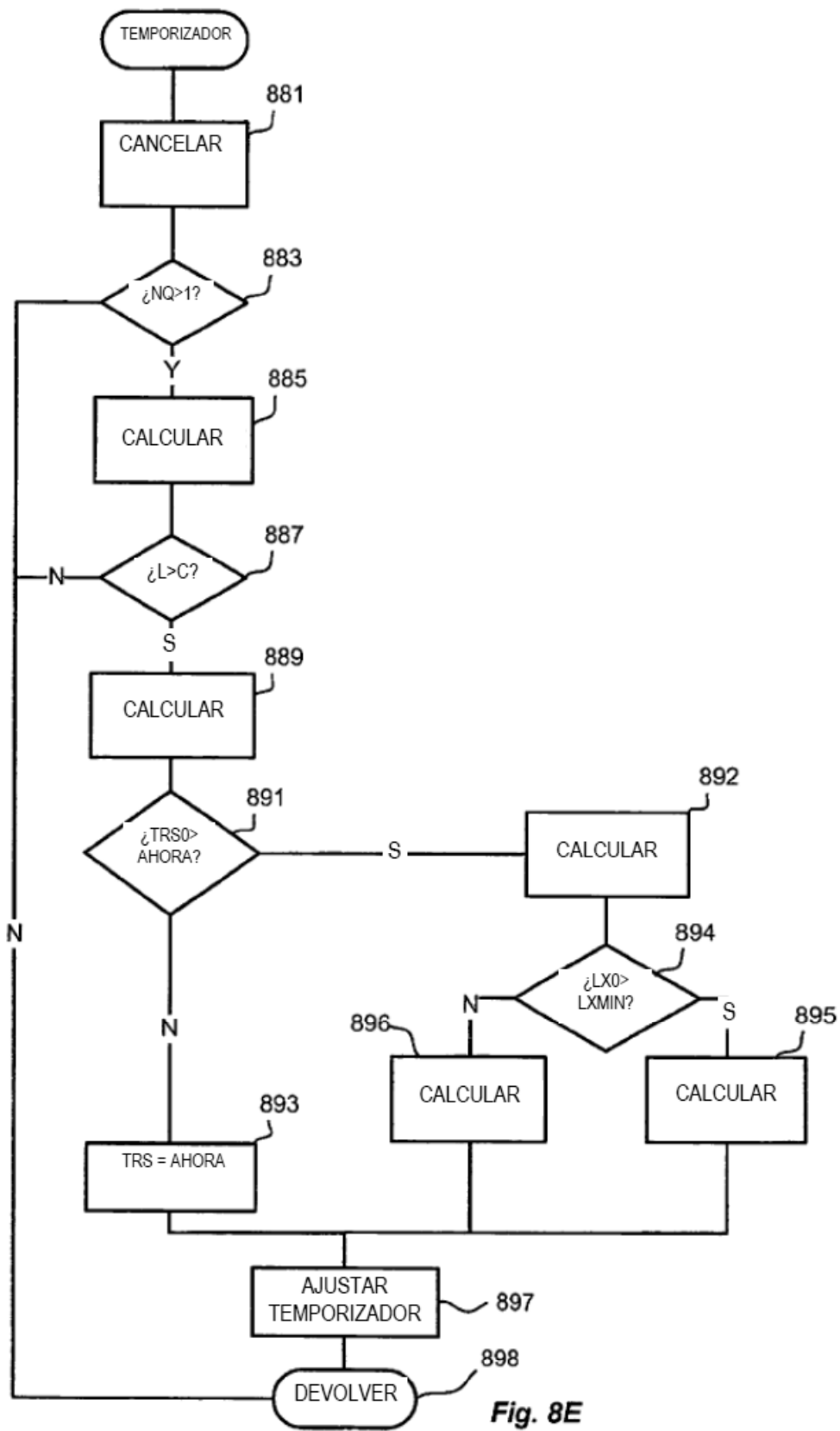


Fig. 8E