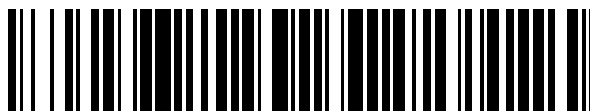


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 379 575**

51 Int. Cl.:

G06F 9/30 (2006.01)

G06F 9/50 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **09700680 .3**

96 Fecha de presentación: **12.01.2009**

97 Número de publicación de la solicitud: **2223203**

97 Fecha de publicación de la solicitud: **01.09.2010**

54 Título: **Realizar un cambio de topología virtual de configuración**

30 Prioridad:
11.01.2008 US 972766

45 Fecha de publicación de la mención BOPI:
27.04.2012

45 Fecha de la publicación del folleto de la patente:
27.04.2012

73 Titular/es:
**INTERNATIONAL BUSINESS MACHINES
CORPORATION
NEW ORCHARD ROAD
ARMONK, NY 10504, US**

72 Inventor/es:
**GAINEY, Charles;
FARRELL, Mark;
KUBALA, Jeffrey y
SCHMIDT, Donald**

74 Agente/Representante:
de Elzaburu Márquez, Alberto

ES 2 379 575 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

Realizar un cambio de topología virtual de configuración.

CAMPO DEL INVENTO

5 El presente invento se refiere en general a la virtualización de sistemas multiprocesadores. En particular, el presente invento se refiere a habilitar programas para cambiar elementos de la topología de su entorno virtual.

ANTECEDENTES

10 Entre las funciones de control del sistema está la capacidad de dividir el sistema en varias particiones lógicas (LPAR). Una LPAR es un subconjunto del hardware del sistema que está definida para soportar un sistema operativo. Una LPAR contiene recursos (procesadores, memoria, y dispositivos de entrada/salida) y opera como un sistema independiente. Pueden existir múltiples particiones lógicas dentro de un sistema de hardware de ordenador central.

En los sistemas de ordenador centrales de IBM incluyendo el S/390®, durante muchos años ha existido un límite de 15 LPAR. Las máquinas más recientes tienen 30 (y potencialmente más). Tales máquinas están ejemplificadas por las de z/Architecture®. La z/Architecture® de IBM está descrita en los Principios de Operación de z/Architecture SA22-7832-05 publicados en Abril de 2007 por IBM.

15 Limitaciones prácticas de tamaño de memoria, disponibilidad de I/O, y potencia de tratamiento disponible limitan usualmente el número de LPAR a ser menor que estos máximos.

20 El hardware y el firmware que proporciona la división en particiones es conocido como PR/SM™ (Recurso de Procesador/Gestor de Sistema). Son las funciones PR/SM las que son usadas para crear y ejecutar las LPAR. Esta diferencia entre PR/SM (un equipo integrado) y las LPAR (el resultado de usar PR/SM) es a menudo ignorada y el término LPAR es usado de forma colectiva para el equipo y sus resultados. Los administradores de sistema asignan partes de memoria a cada LPAR y la memoria no puede ser compartida entre las LPAR. Los administradores pueden asignar procesadores (también conocidos como procesadores centrales (CP) o unidades de tratamiento centrales (CPU) para especificar las LPAR o pueden permitir que los controladores de sistema distribuyan o asignen alguno o todos los procesadores a todas las LPAR usando un algoritmo interno de equilibrio de cargas. Pueden asignarse canales (CHPID) a LPAR específicas o pueden ser compartidos por múltiples LPAR, dependiendo de la naturaleza de los dispositivos en cada canal.

Un sistema con un único procesador (procesador CP) puede tener múltiples LPAR. Un PR/SM tiene un distribuidor interno que puede asignar una parte del procesador a cada LPAR, tanto como un distribuidor de sistema operativo asigna una parte de su tiempo de procesador a cada proceso, recorrido, o tarea.

30 Las especificaciones de control de la división en particiones están parcialmente contenidas en los IOCDS y están parcialmente contenidas en un perfil de sistema. Los IOCDS y el perfil residen ambos en el Elemento de Soporte (SE) que, por ejemplo, es simplemente un ordenador portátil dentro del sistema. El SE puede ser conectado a una o más Consolas de Gestión de Hardware (HMC), que, por ejemplo, son ordenadores personales de sobremesa usados para vigilar y controlar el hardware tales como los microprocesadores centrales. Una HMC es más conveniente para usar que un SE y puede controlar varios ordenadores centrales diferentes.

35 Trabajando desde una HMC (o desde un SE, en circunstancias inusuales), un operador prepara un ordenador central para usar seleccionando y cargando un perfil y un IOCDS. Estos crean las LPAR y configuran los canales con números de dispositivos, asignaciones de LPAR, información de múltiples trayectos, y así sucesivamente. Esto es conocido como una Reposición de Activado (POR). Cargando un perfil e IOCDS diferentes, el operador puede cambiar completamente el número y naturaleza de las LPAR y la apariencia de la configuración de I/O. Sin embargo, hacer esto es usualmente disruptivo para cualesquiera sistemas y aplicaciones operativos en funcionamiento y es por ello raro hacerlo sin planificación previa.

Las particiones lógicas (LPAR) son, en la práctica, equivalentes a ordenadores centrales separados.

45 Cada LPAR ejecuta su propio sistema operativo. Este puede ser cualquier sistema operativo de ordenador central; no hay necesidad de ejecutar z/OS® por ejemplo, en cada LPAR. Los planificadores de instalación pueden elegir compartir dispositivos de I/O a través de varias LPAR, pero esta es una decisión local.

50 El administrador de sistema puede asignar uno o más procesadores de sistema para el uso exclusivo de una LPAR. Alternativamente, el administrador puede permitir que todos los procesadores sean usados en alguna o en todas las LPAR. Aquí, las funciones de control del sistema (a menudo conocidas como microcódigo o firmware) proporcionan un distribuidor para compartir los procesadores entre las LPAR seleccionadas. El administrador puede especificar un número

máximo de procesadores concurrentes que ejecutan en cada LPAR. El administrador puede también proporcionar ponderaciones para LPAR diferentes; por ejemplo, especificar que LPAR1 debería recibir el doble de tiempo de procesador que la LPAR2.

5 El sistema operativo en cada LPAR es inicializado (por ejemplo, IPLed) por separado, tiene su propia copia de su sistema operativo, tiene su propia consola de operador (si es necesario), y así sucesivamente. Si el sistema en una LPAR falla, no tiene efecto sobre las otras LPAR.

10 En un sistema de ordenador central con tres LPAR, por ejemplo, se podría tener una z/OS de producción en LPAR1, una versión de prueba de z/OS en LPAR2, y Linux® para S/390® en LPAR3. Si este sistema total tiene 8 GB de memoria, se podrían haber asignado 4 GB a LPAR1, 1 GB a LPAR2, 1 GB a LPAR3, y haber conservado 2 GB en reserva. Las consolas del sistema operativo para las dos LPAR de z/OS podrían estar en posiciones completamente diferentes.

Para la mayor parte de los propósitos prácticos no hay diferencia entre, por ejemplo, tres ordenadores centrales separados que ejecutan z/OS (y que comparten la mayor parte de su configuración de I/O) y tres LPAR en el mismo ordenador central haciendo las mismas cosas. Con excepciones menores el z/OS, los operadores y las aplicaciones, no pueden detectar la diferencia.

15 Las menores diferencias incluyen la capacidad de z/OS (si se ha permitido cuando las LPAR fueron definidas o en cualquier instante durante la ejecución) para obtener información sobre el rendimiento y la utilización a través del sistema de ordenador central completo y para desplazar dinámicamente recursos (procesadores y canales) entre las LPAR para mejorar el rendimiento.

20 Los ordenadores centrales de IBM® actuales, también llamados un complejo de procesador central (CPC) o complejo electrónico central (CEC), pueden contener varios tipos diferentes de procesadores de z/Architecture® que pueden ser usados para propósitos ligeramente diferentes.

25 Varios de estos propósitos están relacionados con el control de costes de software, mientras que otros son más fundamentales. La totalidad de los procesadores en el CPC comienza como unidades de procesador equivalentes (PU) o máquinas que no han sido caracterizadas para ser usadas. Cada procesador es caracterizado por IBM durante la instalación o en un instante posterior. Las caracterizaciones potenciales son:

- Procesador (CP)

Este tipo de procesador está disponible para el sistema operativo y software de aplicación normales.

- Procesador de Asistencia del Sistema (SAP)

30 Cada ordenador central moderno tiene al menos un SAP; los sistemas mayores pueden tener varios. Los SAP ejecutan código interno para proporcionar el subsistema de I/O. Un SAP, por ejemplo, traduce número de dispositivos y direcciones reales de identificadores de trayecto de canal (CHPID), direcciones de unidad de control, y números dispositivo. Gestiona múltiples trayectos para controlar unidades y realiza la recuperación de error para errores temporales. Los sistemas operativos y las aplicaciones no pueden detectar SAP, y los SAP no usan ninguna memoria "normal".

35 - Instalación Integrada para Linux (IFL)

Este es un procesador normal con una o dos instrucciones inhabilitadas que son usados solamente por z/OS. Linux no usa estas instrucciones y puede por ello operar en un IFL. Linux puede ser ejecutado por un CP también. La diferencia es que en un IFL no es contado cuando se especifica el número de modelo del sistema. Esto puede constituir una diferencia sustancial en costes de software.

40 - zAAP

45 Este es un procesador con varias funciones inhabilitadas (manipulación de interrupción, algunas instrucciones) de tal modo que el sistema operativo completo no puede operar sobre el procesador. Sin embargo, z/OS puede detectar la presencia de procesadores zAAP y los usará para ejecutar el código Java. El mismo código Java puede ser ejecutado sobre un CP estándar. De nuevo, las máquinas zAAP no son contadas cuando se especifica el número de modelo del sistema. Como los IFL, existen solamente para controlar costes de software.

- zIIP

El Procesador de Información Integrada de Sistema z9™ (zIIP) es una máquina especializada para tratar cargas de trabajo de bases de datos que se pueden elegir. El zIIP está diseñado para ayudar a reducir costes de software para seleccionar cargas de trabajo en el ordenador central, tales como inteligencia de negocio (BI),

planificación de recursos de empresa (ERP) y gestión de relaciones con clientes (CRM). El zIIP refuerza la misión de los ordenadores centrales como cubo de datos de la empresa ayudando a hacer un acceso directo a DB2® más efectivo de coste y reduciendo la necesidad de múltiples copias de los datos.

Capacidad de Acoplamiento Integrado (ICF)

- 5 Estos procesadores ejecutan sólo código interno licenciado. No son visibles para sistemas o aplicaciones operativas normales. Por ejemplo, un equipo o instalación de acoplamiento es, en efecto, una gran memoria de apuntes usada por múltiples sistemas para coordinar el trabajo. Los ICF deben ser asignados a las LPAR que entonces resultan equipos de acoplamiento.

Repuestos

- 10 Una PU sin caracterizar funciona como un "repuesto". Si los controladores de sistema detectan una CP o SAP que falla, puede ser reemplazada con una PU de repuesto. En la mayor parte de los casos esto puede hacerse sin ninguna interrupción del sistema, incluso para la aplicación que se ejecuta sobre el procesador que falla. Además de estas caracterizaciones de procesadores, algunos ordenadores centrales tienen modelos o versiones, que están configurados para operar más lentos que la velocidad potencial de sus CP. Esto es ampliamente conocido como "taladrar la rodilla (knee-capping)", aunque IBM prefiere el término ajuste de capacidad, o algo similar. Esto se ha hecho, por ejemplo, usando microcódigo para insertar ciclos nulos en la corriente de instrucciones del procesador. El propósito, de nuevo, es controlar costes de software teniendo el modelo o versión de ordenador central mínimo que satisface los requisitos de la aplicación. Los IFL, SAP, zAAP, zIIP, e ICF siempre funcionan a velocidad total del procesador debido a que estos procesadores "no cuentan" en los cálculos de precio de software.
- 15
- 20 El procesador y la CPU pueden referirse o bien a la caja de sistema completa, o bien a uno de los procesadores (CPU) dentro de la caja del sistema. Aunque el significado puede ser claro a partir del contexto de una descripción, incluso los profesionales de los ordenadores centrales deben clarificar qué procesador o CPU significa que están usando en una descripción. IBM usa el término complejo procesador central (CPC) para referirse a la colección física de hardware que incluye almacenamiento principal, uno o más procesadores centrales, temporizadores, y canales. (Algunos programadores del sistema usan el término complejo electrónico central (CEC) para referirse a la "caja" del ordenador central, pero el término preferido es CPC).
- 25

Brevemente, todos los procesadores S/390 o de z/Architecture dentro de un CPC son unidades de tratamiento (PU). Cuando IBM entrega el CPC, las PU son caracterizadas como CP (para trabajo normal), Equipo Integrado para Linux (IFL), facilidad de acoplamiento integrada (IFC) para configuraciones Sysplex Paralelas, y así sucesivamente.

- 30 Los profesionales de los ordenadores centrales usan típicamente sistema para indicar la caja de hardware, un entorno de hardware completo (con dispositivos de I/O), o un entorno operativo (con software), dependiendo del contexto. Típicamente usan procesador para significar un único procesador (CP) dentro del CPC.

- El z/VM® HYPERVISOR™ está diseñado para ayudar a los clientes a extender el valor de negocio de la tecnología del ordenador central a través de la empresa integrando aplicaciones y datos al tiempo que proporciona niveles excepcionales de disponibilidad, seguridad, y facilidad operativa. La tecnología de virtualización z/VM está diseñada para permitir capacidad para que los clientes hagan funcionar de centenares a millares de servidores Linux en un único ordenador central que funcionan con otros sistemas operativos de Sistema z, tales como z/OS, o como una solución de servidor de empresa a gran escala sólo de Linux. La z/VM V5.3 puede también ayudar a mejorar la productividad acogiendo cargas de trabajo que no son de Linux tales como z/OS, z/VSE, y z/TPF.
- 35

- 40 La z/VM proporciona a cada usuario con un entorno de trabajo individual conocido como una máquina virtual. La máquina en virtual simular la existencia de una máquina real dedicada, incluyendo funciones de procesador, memoria, redes, y recursos de entrada/salida (I/O). Los sistemas operativos y los programas de aplicación pueden ejecutarse en máquinas virtuales como invitados. Por ejemplo, se pueden ejecutar múltiples imágenes de Linux y de z/OS sobre el mismo sistema z/VM que está soportando distintas aplicaciones y usuarios finales. Como resultado, entornos de desarrollo, prueba, y producción pueden compartir un único ordenador físico.
- 45

Con referencia a las figs. 15A -15D, la división en particiones y virtualización implica un cambio en la forma de pensar de físico a lógico tratando recursos de sistema como grupos lógicos en vez de cómo entidades físicas separadas. Esto implica consolidar y agrupar recursos de sistema, y proporcionar una "ilusión de un solo sistema" tanto para servidores homogéneos como heterogéneos, almacenamiento, sistemas distribuidos y redes.

- 50 La división en particiones de hardware implica CPU separadas para sistemas operativos separados, cada una de las cuales ejecuta sus aplicaciones específicas. La división en particiones de software emplea un "hipervisor" basado en software para permitir que los sistemas operativos individuales se ejecuten sobre cualquiera de las CPU o sobre la

totalidad.

Los hipervisores permiten que sistemas operativos múltiples funcionen sobre un ordenador anfitrión al mismo tiempo. La tecnología de hipervisores se ha originado en el IBM VM/370, el predecesor del z/VM que se tiene hoy en día. La división en particiones lógicas (LPAR) implica firmware de división en particiones (un hipervisor basado en hardware, por ejemplo PR/SM) para aislar el sistema operativo de las CPU.

La virtualización permite o explota cuatro capacidades fundamentales: compartir recursos, agregar recursos, emulación de funciones y aislamiento. Se exploran estos tópicos con más detalle en las siguientes secciones.

z/VM es un sistema operativo para la plataforma de Sistema z de IBM que proporciona un entorno de prueba y de producción muy flexibles. La puesta en práctica de z/VM de tecnología de virtualización de IBM proporciona la capacidad de hacer funcionar sistemas operativos de función completa tales como Linux en Sistema z, z/OS, y otros como “invitados” de z/VM. El z/VM soporta invitados de z/Architecture de IBM de 64 bits e invitados de Arquitectura /390 de Sistemas de Empresa de IBM de 31 bits.

El z/VM proporciona a cada usuario con un entorno de trabajo individual conocido como una máquina virtual. La máquina virtual simula la existencia de una máquina real dedicada, incluyendo funciones de procesador, memoria, redes, y recursos de entrada/salida (I/O). Los sistemas operativos y los programas de aplicación pueden funcionar en máquinas virtuales como invitados. Por ejemplo se pueden ejecutar múltiples imágenes de Linux y z/OS sobre el mismo sistema z/VM que está también soportando distintas aplicaciones y usuarios finales. Como resultado, entornos de desarrollo, de pruebas, y de producción pueden compartir un único ordenador físico.

Una máquina virtual usa recursos de hardware reales, pero incluso con dispositivos dedicados (como una unidad de cinta), la dirección virtual de la unidad de cinta puede o no ser la misma que la dirección real de la unidad de cinta. Por ello, una máquina virtual sólo conoce “hardware virtual” que puede o no existir en el mundo real.

Por ejemplo, en un sistema de modo básico, un z/VM de primer nivel es el sistema operativo base que está instalado sobre la parte superior del hardware real fig. 16. Un sistema operativo de segundo nivel es un sistema que está creado sobre el sistema operativo z/VM de base. Por ello, el z/VM como un sistema operativo base funciona sobre el hardware, mientras que un sistema operativo invitado funciona sobre la tecnología de virtualización. En la fig. 14, se ilustra un OS de z/VM invitado de segundo nivel cargado en una partición invitada de primer nivel (invitado-1).

En otras palabras, hay un sistema operativo z/VM de primer nivel que se asienta directamente sobre el hardware, pero los invitados de este sistema z/VM de primer nivel son virtualizados. Virtualizando el hardware desde los invitados, se es capaz de crear y usar tantos invitados como sea necesario con una pequeña cantidad de hardware.

Como se ha mencionado previamente los sistemas operativos que funcionan en máquinas virtuales son a menudo llamados “invitados”. Otros términos y frases que podrían encontrarse son:

- “Funcionamiento de primer nivel” o “funcionamiento de forma nativa” significa que funcionan directamente sobre el hardware (que es lo que el z/VM hace).

- “Funcionamiento de segundo nivel”, “funcionamiento bajo VM”, o “funcionamiento sobre (encima de) VM”, o “funcionamiento como un invitado-1” significa que funcionan como un invitado. Usando el sistema operativo de z/VM, es también posible “funcionar como un invitado-2” cuando el propio z/VM funciona como un invitado-1 sobre un hipervisor PR/SM.

Un ejemplo de la funcionalidad de z/VM es, si se tiene un sistema z/VM de primer nivel y un sistema z/VM de segundo nivel, se podría continuar creando más sistemas operativos sobre el sistema de segundo nivel. Este tipo de entorno es particularmente útil para probar la instalación de un sistema operativo antes del despliegue, o para probar o depurar sistemas operativos. Los recursos virtuales pueden tener funciones o características que no están disponibles en sus recursos físicos subyacentes. La fig. 14 ilustra la virtualización por emulación de recursos. Tales funciones o características se dice que son emuladas por el programa anfitrión de tal modo que el invitado observa la función o característica que ha de ser proporcionada por el sistema cuando es realmente proporcionada debido a la asistencia del programa anfitrión.

Ejemplos incluyen software de emulación de arquitectura que pone en práctica arquitectura de un procesador usando otro; iSCSI, que pone en práctica un bus SCSI virtual sobre una red IP; y almacenamiento en cinta virtual puesta en práctica sobre almacenamiento en disco físico.

Además el empaquetado de unidades de tratamiento central (CPU) en tecnología moderna es a menudo jerárquico. Múltiples núcleos pueden estar situados en un solo chip. Múltiples chips pueden estar situados en un único módulo. Múltiples módulos pueden ser empaquetados sobre una placa, denominada a menudo como un libro, y múltiples libros

pueden ser distribuidos a través de múltiples cuadros.

Las CPU tienen a menudo varios niveles de memorias caché, por ejemplo cada procesador puede tener una caché (o posiblemente una caché de instrucciones y una caché de datos divididas) y puede haber cachés mayores adicionales entre cada procesador y la interfaz de memoria principal. Dependiendo del nivel de la jerarquía, las cachés están también situadas a fin de mejorar el rendimiento total, y a ciertos niveles, una caché puede ser compartida entre más de una sola CPU. Las decisiones de ingeniería relativas a tal colocación se relacionan con el espacio, potencia/térmica, distancias de cableado, frecuencia de CPU, velocidad de memoria, rendimiento del sistema, y otros aspectos. Esta colocación de elementos de la CPU crea una estructura interna que puede ser más o menos favorable a una partición lógica particular, dependiendo de donde reside la situación de cada CPU de la partición. Una partición lógica da la apariencia a un sistema operativo, de propietario de ciertos recursos incluyendo la utilización del procesador en el que en la actualidad, el sistema operativo está compartiendo los recursos con otros sistemas operativos en otras particiones. Normalmente, el software no es conocedor de la situación y, en una configuración de multitratamiento simétrico (SMP), observa un conjunto de CPU donde cada CPU proporciona el mismo nivel de rendimiento. El problema es que la ignorancia del empaquetado interno y la "distancia" entre cualesquiera dos CPU puede dar como resultado que el software lleve a cabo elecciones que no sean óptimas sobre a cuántas CPU puede serles asignado trabajo. Por ello, el potencial completo de la configuración SMP no es conseguido.

El ejemplo de ordenador central de virtualización presentado está destinado a mostrar distintas topologías posibles en la virtualización de una máquina. Como se ha mencionado, los programas que se ejecutan en una partición (incluyendo los sistemas operativos) probablemente tienen una visión de que los recursos disponibles para ellos, incluyendo los procesadores, la memoria y I/O están dedicados a la partición. De hecho, los programas no tienen ninguna idea de que están ejecutándose en una partición. Tales programas tampoco saben de la topología de su partición y por ello no pueden hacer elecciones basadas en tal topología. Lo que se necesita es un medio para que los programas optimicen la topología de configuración sobre la que se están ejecutando.

El documento EP-A-1.647.987 describe un sistema de ordenador anfitrión dividido en particiones lógicamente, que implica un cambio de configuración de una topología de una pluralidad de procesadores invitados de una configuración de invitado.

EXPOSICIÓN DEL INVENTO

El invento proporciona un método según la reivindicación 1 y un sistema y programa de ordenador correspondientes.

Los inconvenientes de la técnica anterior son superados, y se proporcionan ventajas adicionales, mediante la provisión de un método, sistema y producto de programa de ordenador para habilitar un subconjunto de recursos de hardware de ordenador durmientes en un sistema de ordenador actualizable que tiene un conjunto de recursos de hardware de ordenador durmientes.

Un ordenador anfitrión que comprende CPU anfitrionas puede ser dividido en particiones lógicas/virtuales que tienen CPU invitadas. La división en particiones es conseguida preferiblemente por el firmware o por el software como podría ser proporcionado por un sistema operativo tal como z/VM de IBM. Cada CPU invitada es una CPU virtual en que los programas invitados ven las CPU invitadas como procesadores de CPU reales, pero de hecho, el anfitrión subyacente está haciendo corresponder cada CPU invitada a recursos de CPU anfitrionas. En una realización, una CPU invitada es puesta en práctica utilizando una parte de una CPU anfitriona por el anfitrión que designa una parte del tiempo de CPU para la CPU invitada para utilizar la CPU anfitriona. Se ha considerado que una pluralidad de CPU invitadas podría ser soportada por una única CPU anfitriona pero también puede aplicarse lo contrario.

En otra realización, las CPU invitadas son emuladas por el software por el cual, las rutinas de emulación convierten funciones de la CPU invitada (incluyendo decodificación y ejecución de instrucción) a rutinas que funcionan en las CPU anfitrionas. Las CPU anfitrionas están previstas o cubiertas para soportar CPU invitadas. En otra realización, una primera imagen invitada puede ser la anfitriona de una segunda imagen invitada. En cuyo caso las segundas CPU invitadas están cubiertas por las primeras CPU invitadas que son a su vez cubiertas por las CPU anfitrionas. La topología de la configuración es una anidación de niveles de CPU invitadas y uno más CPU anfitrionas.

Una nueva instrucción de REALIZAR EQUIPO O INSTALACIÓN DE TOPOLOGÍA (PTF) es proporcionada y la instrucción de ALMACENAR INFORMACIÓN DEL SISTEMA (STSI) de la técnica anterior es mejorada para proporcionar un nuevo SYSIB (identificador SYSIB 15.1.2) que proporciona información de afinidad de componente y empaquetado lógico al software. Esto permite que el software aplique la selección informada e inteligente sobre cómo los elementos individuales, tales comunidades de tratamiento del multiprocesador, son asignados a distintas aplicaciones y cargas de trabajo. Proporcionando así información a un programa (OS) para mejorar el rendimiento aumentando las relaciones afectadas de memoria caché compartida por ejemplo.

Una nueva instrucción de REALIZAR FUNCIÓN DE TOPOLOGÍA (PTF) es utilizada por un programa privilegiado tal como un supervisor, un OS, un kernel o núcleo y similar para solicitar que la topología de configuración de la CPU dentro de la que el programa se está ejecutando sea cambiada. En una realización, la topología de CPU invitada es conmutada entre la polarización horizontal y vertical.

- 5 Teniendo la capacidad de aprender la información de topología de la CPU, el programa entiende la "distancia" entre cualesquiera dos o más CPU arbitrarias de una configuración de multitratamiento simétrica.

La capacidad proporcionada para minimizar la distancia total de todas las CPU en una configuración, y cómo las tareas de la aplicación de programa particular son distribuidas en la CPU individual proporciona programas supervisores con la habilidad de mejorar el rendimiento. El rendimiento mejorado puede resultar de uno o más de los siguientes atributos que son mejorados por un mejor conocimiento de la topología:

Acortar trayectos de señalización entre las CPU

El almacenamiento compartido, accedido por múltiples CPU es más probablemente que esté en memorias caché que están más próximas al conjunto de las CPU. Por ello, el uso de almacenamiento entre memorias caché es confinado a un subconjunto menor de la máquina total y la configuración que permite una transferencia más rápida de caché a caché. La presencia de una posición de almacenamiento en la memoria caché más próxima de una CPU (L1) es significativamente más probable que ocurra.

Debido al rendimiento mejorado, el número de CPU realmente en la configuración puede ser menor en número, mientras que da aún el mismo trabajo hecho en el mismo o en menor tiempo de ejecución. Tal reducción de CPU reduce el número de trayectos de comunicación que cada CPU debe usar para comunicar con las otras CPU de la configuración, contribuyendo por ello además a una mejora de rendimiento total.

Por ejemplo, si 10 CPU necesitan ejecutar un programa particular, el tráfico entre memorias caché es sustancial mientras que si el mismo programa puede ser ejecutado en una CPU, no hay tráfico entre memorias caché. Esto indica que la presencia en la memoria caché de posiciones de almacenamiento deseadas está garantizada en la memoria caché de la CPU única, si ese almacenamiento está en cualquier memoria caché.

25 Cuando el almacenamiento y la jerarquía de memoria caché asociada es local, en oposición a estar distribuidos a través de múltiples cuadros físicos (es decir cajas, etc.), los trayectos de señalización son más cortos. El conocimiento de topología indica la distancia relativa seleccionando el subconjunto apropiado de CPU para asignar a un programa de aplicación de tal manera que, incluso dentro de un conjunto mayor de CPU en una configuración SMP, el subconjunto optimiza la distancia minimizada entre ellos. Esto es algunas veces llamado un grupo de afinidad.

Las nociones de reducción de cómputo de CPU y distancia entre las CPU son informadas por la información de topología que permite que el programa optimice la asignación de las CPU a un grupo de afinidad.

Un objeto (figs. 20 y 21) del presente invento es proporcionar en un sistema de ordenador anfitrión dividido en particiones lógicamente que comprende procesadores anfitriones (CPU anfitriones), un método, sistema y producto de programa para un cambio de configuración de una topología de una pluralidad de procesadores invitados (CPU invitadas) de una configuración invitada. Preferiblemente, un procesador invitado de la configuración invitada busca una instrucción de REALIZAR FUNCIÓN DE TOPOLOGÍA definida para una arquitectura de ordenador, comprendiendo la instrucción de REALIZAR FUNCIÓN DE TOPOLOGÍA un campo de código de operación que especifica una función que ha de ser realizada. Comprendiendo la función que ha de ser realizada por ejecución de la instrucción de realizar función de topología:

solicitar un cambio especificado de la configuración de la polarización de los procesadores invitados de la configuración invitada; y responder al cambio de polarización especificado solicitado, cambiando la configuración de la topología de los procesadores invitados de la configuración invitada de acuerdo con el cambio de polarización especificado.

45 En un aspecto del invento la instrucción REALIZAR FUNCIÓN DE TOPOLOGÍA comprende además un campo de registro, en que la ejecución de la instrucción de REALIZAR FUNCIÓN DE TOPOLOGÍA comprende además: obtener desde un campo de código de función de un registro especificado por el campo de registro, un valor de campo de código de función, consistiendo el valor del campo de código de función de cualquier especificador de instrucción de polarización horizontal, un especificador de instrucción de polarización vertical o una comprobación del estado de un especificador de cambio de topología; responder a la instrucción que especifica la polarización horizontal, iniciar la polarización horizontal de los procesadores invitados de la configuración de ordenador; responder a la instrucción que especifica la polarización vertical, iniciar la polarización vertical de los procesadores

invitados de la configuración del ordenador; y establecer un valor de código resultante en un campo resultado del registro.

5 En otro aspecto del invento, la polarización iniciada es asíncrona a la finalización de la ejecución, y se comprueba que responde al valor del campo de código de función que especifica una comprobación de estado de un cambio de topología del estado de finalización del cambio de topología.

En una realización, la polarización horizontal comprende proporcionar recursos de procesador anfitrión sustancialmente iguales a cada recurso de procesador invitado, en que la polarización vertical comprende proporcionar sustancialmente más recursos de procesador anfitrión al menos a un procesador invitado de dichos procesadores invitados que al menos a otro procesador invitado de dichos procesadores invitados.

10 En otra realización, el valor de código resultante especifica un código de razón que indica una incapacidad para aceptar la solicitud de polarización y que consiste de: responder a la configuración que es polarizada como es especificado por el código de función antes de la ejecución, indicando el valor de código resultante que la configuración está ya polarizada de acuerdo con el código de función; y, responder a la configuración tratando una polarización incompleta antes de la ejecución, indicando el valor de código resultante que un cambio de topología está ya en proceso.

En una realización, la ejecución comprende además: responder a un cambio de topología que está en progreso, establecer un código de condición que indica un cambio de topología iniciado; y responder a la solicitud que es rechazada, estableciendo un código de condición que indica que la solicitud es rechazada.

20 En una realización, la ejecución comprende además: responder a un informe de que no hay cambio de topología que está pendiente, establecer un código de condición que indica un informe de cambio de topología no pendiente; y responder a un informe de cambio de topología que está pendiente, estableciendo un código de condición que indica un informe de cambio de topología pendiente.

25 En una realización, la instrucción realizar la función de topología definida para la arquitectura de ordenador es buscada y ejecutada por una unidad de tratamiento central de una arquitectura de ordenador alternativa, en que el método comprende además interpretar la instrucción de REALIZAR FUNCIÓN DE TOPOLOGÍA para identificar una rutina de software predeterminada para emular la operación de la instrucción de REALIZAR FUNCIÓN DE TOPOLOGÍA; y en el que ejecutar la instrucción de REALIZAR FUNCIÓN DE TOPOLOGÍA comprende ejecutar la rutina de software predeterminada para realizar operaciones del método para ejecutar la instrucción de máquina.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

30 El sujeto que es considerado como el invento está indicado particularmente y reivindicado distintivamente en la parte de conclusiones de la memoria. El invento, sin embargo, tanto en cuanto a organización como a método de práctica, junto con otros objetos y ventajas del mismo, puede comprenderse mejor por referencia a la siguiente descripción tomada en conexión con los dibujos adjuntos en los que:

La fig. 1 representa un sistema de Ordenador Anfitrión de la técnica anterior;

35 La fig. 2 representa un sistema de Ordenador Anfitrión emulado de la técnica anterior;

La fig. 3 representa un formato de instrucción de una instrucción de máquina STSI de la técnica anterior;

La fig. 4 representa registros implicados de la instrucción STSI de la técnica anterior;

La fig. 5 representa una tabla de código de función;

La fig. 6 representa una tabla SYSIB 1.1.1 de la técnica anterior;

40 La fig. 7 representa una tabla SYSIB 1.2.1 de la técnica anterior;

La fig. 8 representa una tabla SYSIB 1.2.2 de Formato 1 de la técnica anterior;

La fig. 9 representa una tabla SYSIB 1.2.2 de Formato 2 de la técnica anterior;

La fig. 10 representa una tabla SYSIB 15.1.2 de acuerdo con el invento;

La fig. 11 representa una TLE de tipo contenedor;

45 La fig. 12 representa una TLE de tipo CPU;

La fig. 13 representa un formato de instrucción de una instrucción de máquina PTF de acuerdo con el invento;

La fig. 14 representa un formato de registro de la instrucción PTF;

Las figs. 15A-15D representan elementos de la técnica anterior de sistemas de ordenador divididos en particiones;

La fig. 16 representa un sistema de ordenador ejemplar;

5 Las figs. 17-19 representan contenedores del sistema de ordenador ejemplar; y

Las figs. 20-21 representan un flujo de una realización del invento.

DESCRIPCIÓN DETALLADA

10 En un ordenador central, las instrucciones de máquina con arquitectura son usadas por programadores (que escriben típicamente aplicaciones en "C" pero también en Java, COBOL, PL/I, PL/X, Fortran y otros lenguajes de alto nivel), a menudo por medio de una aplicación de compilador. Estas instrucciones almacenadas en el medio de almacenamiento pueden ser ejecutadas de manera nativa en un servidor IBM de z/Architecture, o alternativamente en máquinas que ejecutan otras arquitecturas. Estas pueden ser emuladas en los servidores de ordenador central de IBM existentes y futuros y en otras máquinas de IBM (por ejemplo Servidores de pSeries® y servidores de xSeries®). Pueden ser ejecutadas en máquinas que funcionan con Linux en una amplia variedad de máquinas que
15 usan hardware fabricado por IBM®, Intel®, AMD™, Sun Microsystems y otros. Además de la ejecución en ese hardware bajo una z/Architecture®, Linux puede ser usado así como máquinas que usan emulación por Hercules, UMX, FSI (Fundamental Software, Inc) o Platform Solutions, Inc. (PSI), donde la ejecución generalmente es en un modo de emulación. En el modo de emulación, el software de emulación es ejecutado por un procesador nativo para emular la arquitectura de un procesador emulado.

20 El procesador nativo ejecuta típicamente software de emulación que comprende bien firmware o bien un sistema operativo nativo para realizar la emulación del procesador emulado. El software de emulación es responsable de buscar y ejecutar instrucciones de la arquitectura del procesador emulado. El software de emulación mantiene un contador de programa emulado para mantener la pista de los límites de la instrucción. El software de emulación puede buscar una o más instrucciones de máquina emulada en un instante y convertir la una o más instrucciones de máquina emuladas a un grupo correspondiente de instrucciones de máquina nativa para ejecución por el procesador nativo. Estas instrucciones convertidas pueden ser almacenadas en memoria caché de tal manera que puede ser conseguida una conversión más rápida. De modo general, el software de emulación debe mantener las reglas de arquitectura del la arquitectura de procesador emulado de manera que asegure que los sistemas operativos y aplicaciones escritas para el procesador emulado funcionen correctamente. Además el software de emulación debe
25 proporcionar recursos identificados por la arquitectura de procesador emulado que incluyen, pero no están limitados a registros de control, registros de propósito general (incluyendo a menudo registros de coma flotante), función de traslación de dirección dinámica que incluye tablas de segmentos y tabla de páginas por ejemplo, mecanismos de interrupción, mecanismos de conmutación de contexto, relojes de Hora del Día (TOD) e interfaces con arquitectura a subsistemas I/O de tal modo que un sistema operativo o un programa de aplicación diseñado para funcionar en el procesador emulado, pueden ser ejecutados en el procesador nativo que tiene el software de emulación.
30
35

Una instrucción específica que es emulada es descodificada, y una subrutina es llamada para realizar la función de la instrucción individual. Una función de software de emulación que emula una función de un procesador emulado es implantada, por ejemplo, en una subrutina o unidad "C", o algún otro método de proporcionar una unidad para el hardware específico cómo lo será dentro de los expertos en la técnica después de comprender la descripción de la realización preferida. Distintas patentes de emulación de software y hardware que incluyen, pero no están limitadas al documento US 5551013 para un "Multiprocesador para emulación de hardware" de Beausoleil y col., y al documento US 6009261: "Tratamiento previo de rutinas objetivo almacenadas para emular instrucciones incompatibles en un procesador objetivo" de Scalzi y col.; y al documento US 5574873: Descodificar la instrucción invitada para acceder directamente a las rutinas de emulación que emulan las instrucciones invitadas, de Davidian y col.; al documento US 6308255: Bus de multitratamiento simétrico y conjunto de chips usado para el soporte del coprocesador que permite que un código no nativo se ejecute en un sistema, de Gorishek y col.; y al documento US 6463582: Traductor de código de objeto de optimización dinámica para emulación de arquitectura y método de traducción de código de objeto de optimización dinámica de Lethin y col; y al documento US 5790825: Método para emular instrucciones invitadas en un ordenador anfitrión mediante una nueva compilación dinámica de las instrucciones anfitrionas de Eric Traut; y muchas otras, ilustran la variedad de modos conocidos para conseguir la emulación de un formato de instrucción con arquitectura para una máquina diferente para una máquina objetivo disponible para los expertos en la técnica, así como las técnicas de software comercial usadas por aquellos a que se ha hecho antes referencia.
40
45
50

Con referencia a la fig. 1, los componentes representativos de un sistema de ordenador anfitrión 100 están representados. Pueden también ser empleadas otras disposiciones de componentes en un sistema de ordenador que son bien conocidas en la técnica.

El entorno del ordenador anfitrión está preferiblemente basado de la z/Architecture ofrecida por Internacional Business Machines Corporation (IBM®), Armonk, New York. La z/Architecture está descrita más completamente en: Principios de Operación de la z/Architecture, IBM Pub. N° SA22-7832-05, 6ª, (Abril de 2007). Los entornos informáticos basados en la z/Architecture incluyen, por ejemplo, eServer™ y zSeries®, ambos por IBM®.

El ordenador central 100 representativo comprende una o más CPU 101 en comunicación con almacenamiento principal (memoria de ordenador 102) así como interfaces I/O para dispositivos de almacenamiento 111 y redes 101 para comunicar con otros ordenadores o SAN y similar. La CPU puede tener Traducción de Dirección Dinámica (DAT) 103 para transformar direcciones de programas (direcciones virtuales) en direcciones reales de memoria. Una DAT incluye típicamente una Memoria Tampón de Traducción Anticipada (TLB) 107 para almacenar traducciones caché de manera que los últimos accesos al bloque de memoria del ordenador 102 no requieran el retraso de la traducción de la dirección. Típicamente una memoria caché 109 es empleada entre la memoria de ordenador 102 y el procesador 101. La memoria caché 109 puede ser jerárquica con una gran memoria caché disponible para más de una CPU y memorias caché menores, más rápidas (de menor nivel) entre la gran memoria caché y cada CPU. En alguna puesta en práctica las memorias caché de nivel inferior están divididas para proporcionar memorias caché de bajo nivel separadas para buscar instrucciones y accesos de datos. En una realización, una instrucción es buscada desde la memoria 102 por una unidad de búsqueda de instrucción mediante una memoria caché 109. La instrucción es descodificada en una unidad de descodificación de instrucción (706) y distribuida (con otras instrucciones en algunas realizaciones) a las unidades de ejecución de instrucción 108. Típicamente varias unidades de ejecución 108 son empleadas, por ejemplo una unidad de ejecución aritmética, una unidad de ejecución de coma flotante y una unidad de ejecución de instrucción de derivación. La instrucción es ejecutada por la unidad de ejecución, accediendo a operandos desde los registros de instrucción especificados o desde la memoria según se necesite. Si un operando ha de ser accedido (cargado o almacenado) desde la memoria 102, una unidad 105 de almacenamiento de carga maneja típicamente el acceso bajo control de la instrucción que es ejecutada.

En una realización, el invento puede ser llevado a la práctica mediante software (algunas veces referido como Código Interno con Licencia (LIC), firmware, microcódigo, mili-código, pico-código, y similar, cualquiera de los cuales sería consistente con el presente invento). El código de programa de software que realiza el presente invento es accedido típicamente por el procesador conocido también como una CPU (Unidad de Tratamiento Central) 101 del sistema de ordenador 100 desde medios de almacenamiento 111 a largo plazo, tal como una unidad de CD-ROM, una unidad de cinta o un disco duro. El código del programa de software puede ser realizado en cualquiera de una variedad de medios conocidos para usar con un sistema de tratamiento de datos, tal como un disquete, un disco duro, o un CD-ROM. El código puede ser distribuido sobre tal medio, o puede ser distribuido a los usuarios desde la memoria 102 del ordenador o almacenamiento de un sistema de ordenador sobre una red 110 a otros sistemas de ordenador para utilizar por usuarios de tales otros sistemas.

Alternativamente, el código de programa puede ser realizado en la memoria 102, y accedido por el procesador 101 utilizando el bus o línea de transmisión del procesador. Tal código de programa incluye un sistema operativo que controla la función e interacción de los distintos componentes del ordenador y uno o más programas de aplicación. El código de programa es normalmente paginado desde el medio de almacenamiento denso 111 a la memoria de alta velocidad 102 donde está disponible para el tratamiento por el procesador 101. Las técnicas y métodos para realizar el código de programa de software en memoria, en medio físico, y/o distribuir el código de software mediante redes son bien conocidos y no serán descritos aquí adicionalmente. El código de programa, cuando es creado y almacenado en un medio tangible (que incluye pero no está limitado a módulos de memoria electrónica (RAM), memoria flash, discos compactos (CD), DVD, cinta magnética y similares es a menudo denominado como un "producto de programa de ordenador". El medio del producto de programa de ordenador es típicamente legible por un circuito de tratamiento preferiblemente en un sistema de ordenador para la ejecución por el circuito de tratamiento.

En la fig. 2, está previsto un ejemplo de sistema 201 de ordenador anfitrión emulado que emula un sistema de ordenador central 100 de una arquitectura anfitriona. En el sistema 201 de ordenador anfitrión emulado, el procesador anfitrión (CPU) 208 es un procesador anfitrión emulado (o procesador anfitrión virtual) y comprende un procesador de emulación 207 que tiene una arquitectura de conjunto de instrucciones nativa diferente que la utilizada por el procesador 101 del ordenador anfitrión 100. El sistema 201 de ordenador anfitrión emulado tiene la memoria 202 accesible al procesador de emulación 207. En el ejemplo de realización, la memoria 207 está dividida en particiones en una parte de memoria 102 de ordenador anfitrión y una parte de rutinas de emulación 203. La memoria 102 de ordenador anfitrión está disponible para programas del ordenador anfitrión emulado 201 de acuerdo con la arquitectura de ordenador anfitrión. El procesador de emulación 207 ejecuta instrucciones nativas de un

conjunto de instrucciones con arquitectura de una arquitectura distinta de la del procesador emulado 208, las instrucciones nativas obtenidas desde la memoria de rutinas de emulación 203, y puede acceder a una instrucción anfitriona para la ejecución desde un programa en la memoria 102 de ordenador anfitrión empleando una o más instrucciones obtenidas en una rutina de Secuencia y Acceso/Descodificación que puede descodificar las instrucciones anfitrionas accedidas para determinar una rutina de ejecución de instrucción nativa para emular la función de la instrucción anfitriona accedida.

Otros dispositivos que son definidos para la arquitectura del sistema de ordenador central 100 pueden ser emulados por la Rutina de Instalaciones con Arquitectura, incluyendo dispositivos tales como Registros de Propósito General, Registros de Control, Traducción de Dirección Dinámica, y soporte de Subsistema de I/O y memoria caché del procesador por ejemplo. La rutinas de emulación pueden también aprovecharse de la función disponible en el procesador de emulación 207 (tal como Registros Generales y traducción dinámica de direcciones virtuales) para mejorar el rendimiento de la rutinas de emulación. Un hardware especial y Motores de Descarga pueden también estar previstos para ayudar al procesador 207 en la emulación de la función del ordenador anfitrión 100.

Con el fin de proporcionar información de topología a los programas, son proporcionadas dos instrucciones. La primera es una mejora a una instrucción STSI (Almacenar Información de Sistema) de la técnica anterior y la segunda es una nueva instrucción PTF (Realizar Función de Topología).

Visión general de la Topología de CPU

Con la llegada de los nuevos ordenadores centrales eSeries de IBM, e incluso previamente, la organización de máquinas en estructuras nodales ha dado como resultado en un comportamiento de acceso de memoria no uniforme (NUMA) (algunas veces también llamado "voluminoso"). El propósito de la nueva función SYSIB 15.1.2 de la instrucción STSI (Almacenar Información de Sistema) de la técnica anterior y de la nueva instrucción REALIZAR FUNCIÓN DE TOPOLOGÍA (PTF) es proporcionar conciencia de topología de máquina adicional al programa de manera que ciertas optimizaciones puedan ser realizadas (incluyendo relaciones de memoria caché-éxito mejoradas) y por ello mejorar el rendimiento total. La cantidad de recursos de CPU anfitriona asignada a una configuración invitada de multitratamiento (MP) ha sido generalmente distribuida uniformemente de a través del número de CPU invitadas configuradas. (Una CPU invitada es una CPU lógica proporcionada a un programa, todas las CPU invitadas son soportadas por división en particiones de software/hardware en las CPU anfitrionas reales). Tal distribución uniforme implica que no son autorizadas CPU (o varias CPU) invitadas no particulares a ningún aprovisionamiento de CPU anfitriona extra que cualesquiera otras CPU invitadas determinada arbitrariamente. Esta condición de la configuración invitada, que afecta a todas las CPU de la configuración, es llamada "polarización horizontal". Bajo la polarización horizontal, la asignación de una CPU anfitriona a una CPU invitada es aproximadamente la misma cantidad de aprovisionamiento para cada CPU invitada. Cuando el aprovisionamiento no es dedicado, la misma CPU anfitriona que aprovisiona a la CPU invitada también puede ser utilizada para aprovisionar a la CPU invitada de otro invitado, o incluso a otras CPU invitadas de la misma configuración invitada.

Cuando la otra configuración invitada es una partición lógica diferente, una CPU anfitriona, cuando está activa en cada partición, debe típicamente acceder al almacenamiento principal más debido a que la relación de memoria caché-éxito es reducida teniendo que compartir la memoria caché a través de las zonas de reubicación múltiples. Si el aprovisionamiento de CPU anfitriona puede alterar el balance de tal manera que alguna de las CPU anfitrionas son en su mayor parte, o incluso exclusivamente, asignadas a una configuración invitada dada (y eso resulta el comportamiento normal), entonces las relaciones de memoria caché-éxito mejoran, como lo hace el rendimiento. Tal distribución desigual implica que una o más CPU invitadas están autorizadas al aprovisionamiento de CPU anfitrión extra frente a otras, CPU invitadas determinadas arbitrariamente que están autorizadas a menos aprovisionamiento de CPU anfitriona. Esta condición de la configuración de invitado, que afecta a todas las CPU de la configuración, es llamada "polarización vertical". La arquitectura presentada aquí clasifica la polarización vertical en tres niveles de autorización de aprovisionamiento, alto, medio, y bajo:

- La autorización alta garantiza aproximadamente que el 100% de una CPU anfitriona es asignado a una CPU lógica/virtual, y la afinidad es mantenida como una fuerte correspondencia entre las dos. Con respecto al aprovisionamiento de una partición lógica, cuando la polarización vertical es efectiva, la autorización de una CPU dedicada es definida como alta.

- La autorización media garantiza que una cantidad no específica de recursos de CPU anfitriona (una o más CPU anfitrionas) es asignada a una CPU lógica/virtual, y cualquier capacidad restante de la CPU anfitriona es considerada como un espacio en blanco o vacío que puede ser asignado en otra parte. El mejor caso para el espacio en blanco disponible sería asignarlo como espacio en blanco local si eso es posible. Un resultado menos beneficioso ocurre si el espacio en blanco disponible es asignada como espacio en blanco remoto. Es también el caso de que el porcentaje de recursos asignado a una CPU lógica de la autorización media es una aproximación mucho más suave cuando se compara con la

aproximación del 100% de un conjunto de autorización alta.

- La autorización baja garantiza aproximadamente que el 0% de una CPU anfitriona es asignada a una CPU lógica/virtual. Sin embargo, si hay espacio en blanco disponible, tal CPU lógica/virtual puede recibir aún algún recurso de CPU. Un modelo de contenedores anidados utilizando polarización está destinado a proporcionar un nivel de inteligencia sobre la estructura nodal de la máquina cuando se aplica a la configuración solicitante, de manera que, generalmente, los grupos de CPU anfitrionas pueden ser asignados a grupos de CPU invitadas, mejorando por ello tanto como sea posible el acto de compartir almacenamiento y la minimización de configuraciones deferentes que son esencialmente colindantes en la misma CPU anfitriona. La polarización y autorización indica la relación de las CPU físicas a las CPU lógicas o de las CPU lógicas a las CPU virtuales en una configuración de invitado, y cómo la capacidad asignada a la configuración de invitado es prorrateada a través de las CPU que comprende la configuración.

Históricamente, una configuración invitada ha sido polarizada horizontalmente. Sin embargo muchas CPU invitadas fueron definidas a la configuración, el recurso de CPU anfitriona asignado fue distribuido uniformemente a través de todas las CPU invitadas de una manera equitativa, no autorizada. Puede decirse que el peso de una sola CPU lógica en una partición lógica cuando la polarización horizontal es efectiva es aproximadamente igual al peso de la configuración total dividido por el número de CPU. Sin embargo, con la introducción del 2097 y los modelos de familia, resulta imperativo que se puedan distribuir los recursos de CPU anfitriona de una manera diferente, que es llamada polarización vertical de una configuración, y que a continuación el grado de aprovisionamiento de las CPU invitadas por la CPU anfitriona sea indicado como autorización alta, media o baja. La autorización alta es efectiva cuando una CPU lógica/virtual de una configuración polarizada verticalmente es completamente soportada por la misma CPU anfitriona. La autorización media es efectiva cuando una CPU lógica/virtual de una configuración polarizada verticalmente es parcialmente soportada por una CPU anfitriona. La autorización baja es efectiva cuando a una CPU lógica/virtual de una configuración polarizada verticalmente no se le garantiza ningún recurso de la CPU anfitriona, que no sea el que podría resultar disponible debido al recurso de espacio en blanco que resulta disponible.

Espacio en blanco o vacío de CPU:

Recurso de CPU, hay dos tipos de recursos de espacio en blanco de CPU:

- El espacio en blanco local resulta disponible cuando una CPU lógica/virtual de una configuración no está utilizando todos los recursos a los cuales está autorizada y tal espacio en blanco es a continuación utilizada dentro de la configuración de esa CPU. El espacio en blanco local es preferido con respecto al espacio en blanco remoto cuando se esperan mejores relaciones de memorias caché-éxito cuando el espacio en blanco es utilizado dentro la configuración.
- El espacio en blanco remoto resulta disponible cuando una CPU lógica/virtual de una configuración no está utilizando todos los recursos a los que está autorizada y tal espacio en blanco es a continuación utilizada fuera de la configuración de esa CPU. El espacio en blanco remoto se espera que exhiba menores relaciones de memorias caché-éxito, pero es aún mejor que no ejecute una CPU lógica/virtual en absoluto.

El objetivo es maximizar las relaciones de memoria caché-éxito de la CPU. Para una partición lógica, la cantidad de recursos de CPU física es determinada por ponderaciones del sistema total que determinan el recurso de CPU asignado a cada partición lógica. Por ejemplo, en un MP lógico de 3 vías que es un recurso de CPU física asignado equivalente a una única CPU, y está polarizado horizontalmente, cada CPU lógica sería distribuida independientemente y recibe así aproximadamente el 33% del recurso de CPU física. Si la misma configuración estuviera polarizada verticalmente, solo una sola CPU lógica sería ejecutada y recibiría aproximadamente el 100% del recurso de CPU física (autorización alta) mientras que las dos CPU lógicas restantes no serían distribuidas normalmente (autorización baja). Tal asignación de recursos es normalmente una aproximación. Incluso una CPU de autorización baja puede recibir alguna cantidad de recursos solamente para ayudar a asegurar que un programa no resulta bloqueado en tal CPU. Previendo medios para que un programa de control indique que comprende la polarización, y reciba una indicación para cada CPU de su polarización y, si la polarización es vertical, el grado de autorización, el programa de control puede hacer uso más inteligente de estructuras de datos que están pensadas en general para ser locales a una CPU frente a disponible a todas las CPU de una configuración. También, tal programa de control puede evitar el trabajo directo a cualquier CPU de autorización baja. El recurso real de CPU física asignado podría no constituir un número entero de CPU, así hay también posibilidad de que una o más CPU en una configuración polarizada verticalmente MP que está autorizada pero no a un grado alto, dando como resultado por ello en tales CPU que tienen o bien autorización vertical media o bien baja. Es posible para cualesquiera CPU de autorización baja restantes recibir alguna cantidad de recursos de CPU anfitriona. Por ejemplo, esto puede ocurrir cuando tal CPU es objetivada, tal como mediante una orden SGIP y el recurso de CPU anfitriona de espacio en blanco está disponible. De lo contrario, tal CPU lógica/virtual podría permanecer en un estado sin distribuir, incluso

aunque de otra manera sea capaz de ser distribuida.

Preferiblemente, de acuerdo con el invento, un campo de polarización de 2 bits es definido para la nueva "entrada de lista de topología" (TLE) de tipo CPU de la instrucción ALMACENAR INFORMACIÓN DEL SISTEMA (STSI). El grado de autorización de polarización vertical para cada CPU está indicado como alto, medio o bajo. La asignación no es un porcentaje preciso sino más bien es algo nebuloso y heurístico.

Además de la polarización vertical como medio de reasignar ponderación a las CPU invitadas, existe otro concepto, que es la creación y gestión de capacidad de espacio en blanco (también llamado "espacio en blanco"). La capacidad de espacio en blanco es creada bajo las siguientes circunstancias:

- Una CPU vertical alta contribuye a espacio en blanco cuando su utilización media (AU) cae por debajo del 100 por cien (100-AU).
- Una CPU vertical media que tiene un aprovisionamiento asignado de M por ciento de una CPU anfitriona contribuye a espacio en blanco cuando su utilización media (AU) cae por debajo de M por ciento ($M-AU > 0$).
- Una CPU vertical baja no contribuye a espacio en blanco
- Una CPU vertical alta no es un consumidor de espacio en blanco.

Instrucción de ALMACENAR INFORMACIÓN DE SISTEMA:

Una realización ejemplar de un formato de una instrucción de Almacenar Información de Sistema en la fig. 3 comprende un campo de código de operación 'B27D', un campo de registro base B2 y un campo de desplazamiento con signo D2. El código de operación de la instrucción informa a la máquina que ejecuta la instrucción de que hay implicados registros generales '0' y '1' asociados con la instrucción. Una dirección es obtenida de un segundo operando añadiendo el valor del campo de desplazamiento con signo al contenido del valor de registro general especificado por el campo base. En una realización, cuando el campo del registro base es '0', el valor extendido de signo del campo de desplazamiento es utilizado directamente para especificar el segundo operando. Cuando la instrucción de Almacenar Información de Sistema es buscada y ejecutada, dependiendo de un código de función en el registro general 0, o bien una identificación del nivel de la configuración que ejecuta el programa es situada en el registro general 0 o bien la información acerca de un componente o componentes de una configuración es almacenada en un bloque de información de sistema (SYSIB). Cuando la información acerca de un componente o componentes es solicitada, la información es especificada por otros contenidos del registro general 0 y por contenidos del registro general 1.

El SYSIB, si hay, es designado por la dirección del segundo operando. La máquina se considera que proporciona uno, dos o tres niveles de configuración. Los niveles son:

1. La máquina básica, que es la máquina como si estuviese operando en el modo de ejecución de instrucción básico o no interpretativo.
2. Una partición lógica, que es proporcionada si la máquina está operando en el modo LPAR (dividida en particiones lógicamente, ejecución de instrucción interpretativa). Una partición lógica es proporcionada por el hipervisor LPAR, que es una parte de la máquina. Una máquina básica existe incluso cuando la máquina está operando en el modo LPAR.
3. Una máquina virtual, que es proporcionada por un programa de control de máquina virtual (VM) que es ejecutado bien por la máquina básica o bien en una partición lógica. Una máquina virtual puede ejecutar ella misma un programa de control de VM que proporciona una máquina virtual de nivel más alto (más eliminada de la máquina básica), que es considerada también una configuración de nivel 3. Los términos modo básico, modo LPAR, partición lógica, hipervisor, y máquina virtual, y cualesquiera otros términos relacionados específicamente a esos términos, no son definidos en esta publicación; están definidos en los manuales de máquina. Un programa que es ejecutado por una configuración de nivel 1 (la máquina básica) puede solicitar información acerca de esa configuración. Un programa que es ejecutado por una configuración de nivel 2 (en una partición lógica) puede solicitar información acerca de la partición lógica y acerca de la máquina básica subyacente. Un programa que es ejecutado por una configuración de nivel 3 (una máquina virtual) puede solicitar información acerca de la máquina virtual y acerca de uno o dos niveles subyacentes; una máquina básica es siempre subyacente, y una partición lógica puede o no estar entre la máquina básica y la máquina virtual. Cuando se solicita la información acerca de una máquina virtual, es proporcionada información acerca de la configuración que ejecuta el programa y acerca de cualquier nivel o niveles subyacentes de máquina virtual. En cualquiera de estos

casos, es proporcionada información acerca de un nivel solamente si el nivel pone en práctica la instrucción.

El código de función que determina la operación es un entero binario sin signo en las posiciones 32-35 de bit del registro general 0 y es como sigue:

5 Código de

Función	Información solicitada
0	Número de nivel de configuración actual
1	Información acerca de nivel 1 (la máquina básica)
2	Información acerca de nivel 2 (una partición lógica)
10 3	Información acerca de nivel 3 (una máquina virtual)
4-14	Ninguna; los códigos están reservados
15	Información de nivel de configuración actual

Código de Función Inválido:

15 El nivel de la configuración que ejecuta el programa es denominado el nivel actual. El nivel de configuración especificado por un código de función distinto de cero es llamado el nivel especificado. Cuando el nivel especificado tiene un número más elevado que el nivel actual, entonces el código de función es llamado inválido, el código de función es establecido a 3, y no se realiza ninguna otra acción (incluyendo comprobación).

Código de Función Válido:

20 Cuando el código de función es igual o menor que el número de nivel actual, es llamado válido. En este caso los bits 36-55 del registro general 0 y los bits 32-47 del registro general 1 deben ser cero o 15; de lo contrario, se reconoce una excepción de especificación. Los bits 0-31 de los registros generales 0 y 1 son siempre ignorados. Cuando el código de función es 0, un entero binario sin signo que identifica el nivel de configuración actual (1 para máquina básica, 2 para partición lógica, o 3 para máquina virtual) es situado en posiciones 32-35 de bit del registro general 0, el código de condición es establecido a 0, y no se realiza ninguna otra acción. Cuando el código de función es válido y distinto de cero, los registros generales 0 y 1 contienen especificaciones adicionales acerca de la información solicitada como sigue:

- Las posiciones de bit 56-63 del registro general 0 contiene un entero binario sin signo, llamado selector 1, que especifica un componente o componentes de la configuración especificada.
- Las posiciones de bit 48-63 del registro general 1 contiene un entero binario sin signo, llamado selector 2, que especifica el tipo de información solicitada.

El contenido de los registros generales 0 y 1 está mostrado en la fig. 4.

35 Cuando el código de función es válido y distinto de cero, la información puede ser almacenada en un bloque de información de sistema (SYSIB) que comienza en la ubicación del segundo operando. El SYSIB es 4k bytes y debe comenzar en un límite de 4k bytes; de lo contrario, puede ser reconocida una excepción de especificación, dependiendo del selector 1 y del selector 2 y de si las excepciones de acceso son reconocidas debido a referencias al SYSIB.

40 El selector 1 puede tener valores siguientes:

Selector 1	Información solicitada
0	Ninguna; el selector está reservado

ES 2 379 575 T3

- 1 Información acerca del nivel de configuración especificado por el código de función
- 2 Información acerca de una o más CPU en el nivel de configuración especificado
- 3-255 Ninguna; los selectores están reservados

Cuando el selector 1 es 1, el selector 2 puede tener los valores siguientes:

5 Selector 2 cuando

Selector 1 es 1 Información Solicitada

- 0 Ninguna; el selector está reservado
- 1 Información acerca del nivel de configuración especificado
- 2 Información de topología acerca del nivel de configuración especificado

10 3-65.535 Ninguna; los selectores están reservados

Cuando el selector 1 es 2, el selector 2 tiene los valores siguientes:

Selector 2 cuando

Selector 1 es 2 Información solicitada

- 0 Ninguna; el selector está reservado
- 15 1 Información acerca de la CPU que ejecuta el programa en el nivel de configuración especificado
- 2 Información acerca de todas las CPU en el nivel de configuración especificado
- 3-65.535 Ninguna; los selectores están reservados

20 Solamente ciertas combinaciones del código de función, selector 1, y selector 2 son válidas, como se ha mostrado en la fig. 5.

Cuando la combinación del código de función especificado, selector 1, y selector 2 es inválida (es distinta de la mostrada en la fig. 5), o si es válida pero la información solicitada no está disponible ya que el nivel especificado no pone en práctica o no pone en práctica completamente la instrucción o porque una parte necesaria del nivel no está instalada o no está inicializada, y dado que una excepción no es reconocida, el código de condición es establecido a 3. Cuando el código de función es distinto de cero, la combinación es válida, la información está disponible, y no hay excepción, la información solicitada es almacenada en un bloque de información del sistema (SYSIB) en la dirección del segundo operando.

Alguno o todos los del SYSIB pueden ser buscados antes de ser almacenados.

30 Un SYSIB puede ser identificado en las referencias por medio del "SYSIB fc.s1.s2", donde "fc", "s1", y "s2" son los valores de un código de función, del selector 1 y del selector 2, respectivamente.

Las secciones siguientes describen los SYSIB definidos por medio de cifras y texto relacionado. En las figuras, los desplazamientos mostrados a la izquierda son valores de palabra (comprendiendo una palabra 4 bytes). "La configuración" se refiere al nivel de configuración especificado por el código de función (el nivel de configuración acerca del cual es solicitada información).

35 SYSIB 1.1.1 (Configuración de Máquina Básica)

SYSIB 1.1.1 tiene el formato mostrado en la fig. 6, donde los campos tienen los siguientes significados:

40 Reservado: El contenido de las palabras 0-7, 13-15, y 29-63 están reservados y son almacenados como ceros. Los contenidos de las palabras 64-1023 están reservados y pueden ser almacenados como ceros o pueden permanecer sin cambios.

Fabricante: Las palabras 8-11 contienen el nombre del fabricante de la configuración EBCDIC de 16 caracteres (0-9 o A-Z mayúsculas). El nombre es justificado a la izquierda con espacios en blanco al final si es necesario.

5 Tipo: La palabra 12 contiene el número de tipo de la configuración EBCDIC de cuatro caracteres (0-9). (Esto es llamado el número de tipo máquina en la definición de ALMACENAR ID de CPU).

Identificador de Capacidad de Modelo: Las palabras 16-19 contienen el identificador de la configuración de capacidad de modelo EBCDIC de 16 caracteres (0-9 o A-Z mayúsculas). El identificador de capacidad de modelo es justificado a la izquierda con espacios en blanco al final si es necesario.

10 Código de Secuencia: Las palabras 20-23 contienen el código de secuencia de la configuración EBCDIC de 16 caracteres (0-9 o A-Z mayúsculas). El código de secuencia es justificado a la derecha con ceros que preceden a EBCDIC si es necesario.

Planta de Fabricación: La palabra 24 contiene el código EBCDIC de cuatro caracteres (0-9 o A-Z mayúsculas) que identifica la planta de fabricación para la configuración. El código es justificado a la izquierda con espacios en blanco posteriores si es necesario.

15 Modelo: Cuando la palabra 25 no son ceros binarios, las palabras 25-28 contienen la identificación de modelo de la configuración EBCDIC de 16 caracteres (0-9 o A-Z mayúsculas). La identificación del modelo es justificada a la izquierda con espacios en blanco posteriores si es necesario. (Esto es llamado el número de modelo en nota de programación 4 en la página 10-111 de ALMACENAR ID de CPU). Cuando la palabra 25 son ceros binarios, el contenido de las palabras 16-19 representa tanto el identificador de capacidad de modelo como el modelo.

Notas de programación:

1. Los campos del SYSIB 1.1.1 son similares a los del descriptor de nodos descrito en la publicación Comandos Comunes de Dispositivo I/O y Auto Descripción. Sin embargo, el contenidos de los campos SYSIB puede no ser idéntico al contenido de los campos del descriptor de nodo correspondiente porque los campos SYSIB:

- 25 • Permiten más caracteres.
- Son más flexibles con respecto al tipo de caracteres permitidos.
- Proporcionan información que es justificada de manera diferente dentro del campo.
- Pueden no usar el mismo método para determinar el contenido de los campos tales como el campo del código de secuencia.

30 2. El campo del modelo en un descriptor de nodos corresponde al contenido del campo del modelo STSI y no al campo del identificador de capacidad de modelo STSI.

3. El campo del modelo especifica el modelo de la máquina (es decir el modelo físico); el campo del identificador de capacidad del modelo especifica una señal que puede ser usada para localizar una declaración de capacidad o rendimiento en la publicación de Librería del Sistema para el modelo.

35 SYSIB 1.2.1 (CPU de Máquina Básica)

SYSIB 1.2.1 tiene el formato mostrado en la fig. 7 donde los campos tienen el siguiente significado:

Reservado: El contenido de las palabras 0-19, los bytes 0 y 1 de la palabra 25, y las palabras 26-63 está reservado y es almacenado como ceros. El contenido de las palabras 64-1023 está reservado y puede ser almacenado como ceros o puede permanecer sin cambios.

40 Código de Secuencia: Las palabras 20-23 contienen el código de secuencia de la configuración EBCDIC de 16 caracteres (0-9 o A-Z mayúsculas). El código es justificado a la derecha con ceros precediendo a EBCDIC si es necesario.

45 Planta de fabricación: La palabra 24 contiene el código EBCDIC de cuatro caracteres (0-9 o A-Z mayúsculas) que identifica la planta de fabricación para la configuración. El código es justificado a la izquierda con espacios en blanco posteriores si es necesario.

Dirección CPU: Los Bytes 2 y 3 de la palabra 25 contienen la dirección de la CPU por la que esta CPU es

identificada en una configuración multitratamiento. La dirección de la CPU es un entero binario sin signo de 16 bit. La dirección de la CPU es la misma que la que es almacenada por ALMACENAR DIRECCIÓN DE LA CPU cuando el programa es ejecutado por una máquina que opera en el modo básico.

Nota de Programación:

- 5 Múltiples CPU en la misma configuración tienen el mismo código de secuencia, y es necesario utilizar otra información, tal como la dirección de la CPU, para establecer una identidad única de la CPU. El código de secuencia devuelto para una CPU de máquina básica y una CPU de partición lógica son idénticos y tienen el mismo valor que el código de secuencia devuelto para la configuración de la máquina básica.

SYSIB 1.2.2 (CPU de Máquina Básica)

- 10 El campo de formato en byte 0 de la palabra 0 determina el formato del SYSIB. Cuando el campo de formato tiene un valor de cero, SYSIB 1.2.2 tiene una implantación de formato 0 como se ha mostrado en la fig. 8. Cuando el campo del formato tiene un valor de uno, SYSIB 1.2.2 tiene una implantación de formato 1 como se ha mostrado en la fig. 9.

Reservado:

- 15 Cuando el campo de formato contiene un valor de cero, el contenido de los bytes 1-3 de la palabra 0 y las palabras 1-6 está reservado y almacenado como ceros. Cuando el campo del formato contiene un valor de uno, el contenido del byte 1 de la palabra 0 y las palabras 1-6 está reservado y almacenado como ceros. Cuando son necesarias poco más que 64 palabras para contener la información para todas las CPU, la parte del SYSIB siguiente a la lista del factor de ajuste en un SYSIB de formato 0 o la lista del factor de
 20 ajuste alternativo en un SYSIB de formato 1, hasta la palabra 63 son reservadas y almacenadas como ceros. El contenido de las palabras 64-1023 es reservado y puede ser almacenado como ceros o puede permanecer sin cambios. Cuando son necesarias 64 o más palabras para contener la información para todas las CPU, la parte del SYSIB siguiente a la lista del factor de ajuste en un SYSIB de formato 0 o la lista del factor de ajuste alternativo en un SYSIB de formato 1, hasta la palabra 1023 son reservadas y pueden
 25 ser almacenadas como ceros o permanecer sin cambios.

Formato:

El byte 0 de la palabra 0 contiene un entero binario sin signo de 8 bits que especifica el formato del SYSIB 1.2.2.

Compensación de Capacidad de CPU Alternativa:

- 30 Cuando el campo formato tiene un valor de uno, los bytes 2-3 de la palabra 0 contienen un entero binario sin signo de 16 bits que especifica la compensación en bytes del campo de capacidad de CPU alternativa en el SYSIB.

Capacidad de CPU secundaria:

- 35 La palabra 7 contiene un entero binario sin signo de 32 bits que, cuando no es cero, especifica una capacidad secundaria que puede ser aplicada a ciertos tipos de CPU en la configuración. No hay descripción formal del algoritmo utilizado para generar este entero, excepto que es el mismo que el algoritmo utilizado para generar la capacidad de CPU. El entero es utilizado como una indicación de la capacidad de una CPU con relación a la capacidad de otros modelos de CPU, y también con relación a la capacidad de otros tipos de CPU dentro de un modelo. El valor de la capacidad se aplica a cada una de las CPU de uno o más tipos de CPU aplicables en la configuración. Es decir, todas las CPU en la configuración de un tipo o tipos aplicables tienen la misma capacidad. Cuando el valor es cero, todas las CPU de cualquier tipo de CPU en la configuración tienen la misma capacidad,
 40 como es especificado por la capacidad de la CPU. La capacidad de CPU secundaria puede o no ser el mismo valor que el valor de capacidad de la CPU. Los factores de ajuste de capacidad de la CPU de multitratamiento son también aplicables a las CPU cuya capacidad es especificada por la capacidad de CPU secundaria.

Capacidad de CPU:

- 45 Si los bits 0-8 de la palabra 8 son cero, la palabra contiene un entero binario sin signo de 32 bits (I) en el intervalo $0 < I < 223$ que especifica la capacidad de una de las CPU en la configuración. Si los bits 0-8 de la palabra 8 son distintos de cero, la palabra contiene un número de formato corto de coma flotante binario de 32 bits en vez de un entero binario sin signo de 32 bits. Independientemente de la codificación, un valor inferior indica una capacidad de CPU proporcionalmente más elevada. Más allá de eso, no hay descripción formal del algoritmo utilizado para generar este valor. El valor es utilizado como una indicación de la capacidad de la CPU relativa a la capacidad de otros
 50 modelos de CPU. El valor de capacidad se aplica a cada una de las CPU no secundarias en la configuración. Es

decir, todas las CPU no secundarias en la configuración tienen la misma capacidad.

Cómputo Total de CPU:

Los bytes 0 y 1 de la palabra 9 contienen un entero binario sin signo de 16 bits que especifica el número total de CPU en la configuración. Este número incluye todas las CPU en el estado configurado, el estado en espera, o el estado reservado.

Cómputo de CPU Configuradas:

Los bytes 2 y 3 de la palabra 9 contienen un entero binario sin signo de 16 bits que especifica el número de CPU que están en el estado configurado. Una CPU está en el estado configurado cuando está en la configuración y disponible para ser utilizada para ejecutar programas.

Cómputo de CPU en espera:

Los bytes 0 y 1 de la palabra 10 contienen un número entero sin signo de 16 bits que especifica el número de CPU que están en estado de espera. Una CPU está en estado de espera cuando está en la configuración, no está disponible para ser utilizada para ejecutar programas, y puede pasar a estar disponible emitiendo instrucciones para colocarla en el estado configurado.

Cómputo de CPU Reservadas:

Los bytes 2 y 3 de la palabra 10 contienen un entero binario sin signo de 16 bits que especifica el número de CPU que están en el estado reservado. Una CPU está en el estado reservado cuando está en la configuración, no está disponible para ser utilizada para ejecutar programas, y no puede pasar a estar disponible emitiendo instrucciones para colocarla en el estado configurado. (Puede ser posible colocar una CPU reservada en el estado en espera o configurado por medio de acciones manuales).

Factores de Ajuste de Capacidad de CPU de Multitratamiento

Comenzando con bytes 0 y 1 de palabra 11, el SYSIB contiene una serie de campos de dos bytes contiguos, cada uno de los cuales contiene un entero binario sin signo de 16 bits utilizado para formar un factor de ajuste (fracción) para el valor contenido en el campo de capacidad de CPU. Tal fracción es desarrollada utilizando el valor (V) del primer campo de dos bytes de acuerdo con uno de los siguientes métodos:

- Si V es del orden de $0 < V \leq 100$, se indica un denominador de 100 que produce una fracción de $V/100$.
- Si V es del orden de $101 \leq V < 255$, se indica un denominador de 255 que produce una fracción de $V/255$.
- Si V es del orden de $255 \leq V \leq 65.536$, se indica un denominador de 65.536 que produce una fracción de $V/65.536$.

Así, la fracción representada por cada campo de dos bytes es a continuación desarrollada dividiendo el contenido de un campo de dos bytes por el denominador indicado. El número de campos de factor de ajuste es inferior en uno al número de CPU especificadas en el campo de cómputo total de CPU. Los campos de factor de ajuste corresponden a configuraciones con números crecientes de CPU en el estado configurado. El primer campo de factor de ajuste corresponde a una configuración con dos CPU en el estado configurado. Cada campo de factor de ajuste sucesivo corresponde a una configuración con un número de CPU en el estado configurado que es uno más que para el campo precedente.

Capacidad de CPU Alternativa:

Cuando el campo de formato tiene un valor de uno, si los bits 0-8 de la palabra N son cero, la palabra contiene un entero (I) binario sin signo de 32 bits del orden de $0 \leq I < 223$ que especifica la capacidad anunciada de una de las CPU en la configuración. Si los bits 0-8 de la palabra N son distintos de cero, la palabra contiene un número de formato corto, de coma flotante binario de 32 bits en lugar de un entero binario sin signo de 32 bits. Independientemente de la codificación, un valor inferior indica una capacidad de CPU proporcionalmente mayor. Más allá de eso, no hay descripción formal del algoritmo usado para generar este valor. El valor es usado como una indicación de la capacidad anunciada de la CPU con relación a la capacidad anunciada de otros modelos de CPU. El valor de capacidad alternativa se aplica a cada una de las CPU en la configuración. Es decir, todas las CPU en la configuración tienen la misma capacidad alternativa.

Factores de Ajuste de la Capacidad de CPU Alternativa de Multitratamiento:

Comenzando con los bytes 0 y 1 de la palabra N+1, el SYSIB contiene una serie de campos de dos bytes contiguos, cada

uno de los cuales contiene un entero binario sin signo de 16 bits usado para formar un factor de ajuste (fracción) para el valor contenido en el campo de capacidad de CPU alternativa. Tal fracción es desarrollada usando el valor (V) del primer campo de dos bytes de acuerdo con uno de los siguientes métodos:

- Si V es del orden de $0 < V \leq 100$, se indica un denominador de 100 que produce una fracción de $V/100$.
- Si V es del orden de $101 \leq V < 255$, se indica un denominador de 255 que produce una fracción de $V/255$.
- Si V es del orden de $255 \leq V \leq 65.536$, se indica un denominador de 65.536 que produce una fracción de $V/65.536$.

Así, la fracción representada por cada campo de dos bytes es a continuación desarrollada dividiendo el contenido de un campo de dos bytes por el denominador indicado. El número de campos de factor de ajuste alternativo es inferior en uno al número de CPU especificadas en el campo de cómputo de CPU totales. Los campos de factor de ajuste alternativo corresponden a configuraciones con números crecientes de CPU en el estado configurado. El primer campo de factor de ajuste alternativo corresponde a una configuración con dos CPU en el estado configurado. Cada campo de factor de ajuste alternativo sucesivo corresponde a una configuración con un número de CPU en el estado configurado que es uno más que para el campo precedente.

SISIB 15.1.2 (Topología de Configuración)

SYSIB 15.1.2 tiene el formato mostrado en la fig. 10. Los campos tienen los significados siguientes:

Reservado: El contenido de 0-1 bytes de la palabra 0, el byte 2 de palabra 2, y la palabra 3 es reservado y almacenado como ceros. El contenido de palabras N-1023 es reservado y puede ser almacenado como ceros o puede permanecer sin cambios.

Longitud: Los bytes 2-3 de la palabra 0 contienen un entero binario sin signo de 16 bits cuyo valor es el cómputo de bytes del entero SYSIB 15.1.2. La longitud solo de la lista de topología es determinada sustrayendo 16 del valor de longitud en los bytes 2-3 de la palabra 0. N en la fig. 10 es determinado evaluando la fórmula $N = \text{Longitud}/4$.

Mag1-6: La palabra y bytes 0-1 de la palabra constituyen seis campos de un byte en que el contenido de cada byte indica el número máximo de entradas de lista de topología (TLE) de tipo contenedor o las TLE de tipo CPU en el nivel de anidación correspondiente. Las TLE de tipo CPU se han encontrado siempre solamente en el nivel Mag1. Adicionalmente, el valor Mag1 también especifica el número máximo de CPU que pueden estar representadas por una TLE de tipo contenedor del nivel Mag2. Cuando el valor del nivel de anidación es mayor que uno, los niveles de anidación de contención por encima del nivel Mag1 están ocupados solamente por las TLE de tipo contenedor. Un cambio dinámico en la topología puede alterar el número de TLE y el número de CPU en el nivel Mag1, pero los límites representados por los valores de los campos Mag1-6 no cambian dentro de una familia de máquinas de modelo.

La topología es una lista de entradas estructurada en la que una entrada define una o más CPU u otras están implicadas con la estructura de anidación. Lo siguiente ilustra el significado de los campos de magnitud:

Cuando todas las CPU de la máquina son pares y no existe organización de contención distinta de la totalidad del propio complejo de tratamiento central, el valor del nivel de anidación es 1, Mag1 es el único campo de magnitud distinto de cero, y el número de TLE de tipo CPU almacenado no excede del valor de Mag1.

Cuando todas las CPU de la máquina están subdivididas en grupos de pares de tal modo que existe un nivel de contención, el valor del nivel de anidación es 2, Mag1 y Mag2 son los únicos campos de magnitud distinta de cero, el número de TLE de tipo contenedor almacenadas no excede del valor de Mag2, y el número de TLE de tipo CPU almacenadas dentro de cada contenedor no excede del valor de Mag1.

Los bytes Mag3-6 resultan usados de manera similar (prosiguiendo en una dirección de derecha a izquierda) cuando el valor del nivel de anidación cae al intervalo 3-6.

Mnest: El byte 3 de la palabra 2 especifica el nivel de anidación de la topología a la que puede ser extendida la configuración al tiempo que continúa permitiendo que el programa invitado sobreviva. El máximo valor Mnest es seis: el mínimo es uno. Si Mnest es uno, no hay estructura de anidación de TLE real, Mag1 es el único campo distinto de cero en el intervalo Mag1-6, y solo las TLE de tipo CPU están representadas en la lista de topología. El valor Mnest representa el número de valores de magnitud distinta de cero que comienza con el campo de magnitud en el byte de la palabra 2 (Mag1), prosiguiendo a la izquierda cuando Mnest es mayor que uno, y con los campos de magnitud restantes almacenados como ceros.

El valor de Mnest es la anidación máxima posible. Ningún cambio de configuración dinámico excede de este límite.

Lista de topología: Las palabras de la fig. 10 en el margen 4 a N-1 especifican una lista de una o más entradas de lista de topología (TLE). Cada TLE es un campo de ocho bytes o de dieciséis bytes; así N es un número par de palabras, y un corolario es que una TLE siempre comienza en un límite de doble palabra.

- 5 Entradas de Lista de Topología: La primera TLE en la lista de topología comienza en un nivel de anidación igual a MNest-1. La lista de topología completa representa la configuración del emisor de la instrucción STSI que especifica SYSIB 15.1.2; ninguna entrada de TLE de contenedor más exterior es utilizada ya que sería redundante con la lista completa, y la configuración completa. Por ello, el mayor nivel de anidación puede tener más de un único contenedor de pares.

Las figs. 11 y 12 ilustran los tipos de TLE, en que los campos tienen las siguientes definiciones:

- 10 Nivel de Anidación (NL): Byte 0 de palabra 0 especifica en nivel de anidación de TLE.

NL	Significado
0	La TLE es una TLE de tipo CPU
1 – 5	La TLE es una TLE de tipo contenedor. La primera TLE de tipo contenedor almacenada en una lista de topología o un recipiente padre tiene un ID de contenedor del orden de 1-255. Si existen contenedores hermanos dentro del mismo padre, prosiguen en orden ascendente de ID de contenedor, que puede o no ser consecutivo, a un valor máximo de 255.
06-FF	Reservado.

- 20 Las TLE hermanas tienen el mismo valor de nivel de anidación que es equivalente bien al valor del nivel de anidación menos uno de la TLE padre inmediato, o al valor de MNest menos uno, debido a que el padre inmediato es la lista de topología en vez de una TLE.

Reservado 0: Para una TLE de tipo contenedor, los bytes -3 de la palabra 0 y los bytes 0-2 de la palabra 1 son reservados y almacenados como ceros. Para una TLE de tipo CPU, los bytes 1-3 de la palabra 0 y los bits 0-4 de la palabra 1 son reservados y almacenados como ceros.

- 25 ID de Contenedor:

El byte de la palabra 1 de una TLE de tipo contenedor especifica un entero binario distinto de cero sin signo de 8 bits cuyo valor es el identificador del contenedor. El ID de contenedor para una TLE es único dentro del mismo contenedor padre.

Dedicada (D):

- 30 El byte 5 de la palabra 1 de una TLE de tipo CPU, cuando es uno, indica que una o más CPU representadas por la TLE son dedicadas. Cuando D es cero, una o más CPU de la TLE son no dedicadas.

Polarización (PP):

Los bytes 6-7 de la palabra 1 de una TLE de tipo CPU especifica el valor de polarización y, cuando la polarización es vertical, el grado de polarización vertical también llamado autorización (alta, media, baja) de las CPU correspondientes representadas por la TLE. Se usan los valores siguientes:

- 35 Significado de PP:

- 0 La una o más CPU representadas por la TLE están polarizadas horizontalmente
- 1 La una o más CPU representadas por la TLE están polarizadas verticalmente. La autorización es baja.
- 2 La una o más CPU representadas por la TLE están polarizadas verticalmente. La autorización es media.
- 3 La una o más CPU representadas por el TLE están polarizadas verticalmente. La autorización es alta.

- 40 La polarización es sólo significativa en una configuración de multitratamiento lógica y virtual que utiliza procesadores anfitriones compartidos y accede a cómo es aplicado el recurso asignado a una configuración a través de las CPU de la configuración. Cuando la polarización es efectiva, a cada CPU de una configuración se le garantiza aproximadamente la misma cantidad de recursos. Cuando la polarización vertical es efectiva, las CPU de una configuración son clasificadas en

tres niveles de autorización de recursos: alto, medio, y bajo.

5 Tanto la reposición del subsistema como la ejecución satisfactoria de la orden de arquitectura establecida SIGO que especifica el modo ESA/390 sitúa una configuración de todas sus CPU en polarización horizontal. Las CPU inmediatamente afectadas son aquellas que están en el estado configurado. Cuando una CPU en el estado de espera está configurada, adquiere la polarización común de la configuración y hace que un cambio de topología de esa configuración sea reconocido.

Una CPU dedicada está o bien polarizada horizontalmente o verticalmente. Cuando una CPU dedicada está polarizada verticalmente, la autorización es siempre alta. Así cuando D es uno, PP es o bien un 00 binario o bien un 11 binario.

Tipo CPU:

10 El byte 1 de la palabra 1 de una TLE de tipo CPU especifica un entero binario sin signo de 8 bits cuyo valor es el tipo de CPU de la una o más CPU representadas por la TLE. El valor de tipo CPU especifica bien un tipo CPU primario o cualquiera de los posibles tipos de CPU secundarios.

Origen de Dirección de CPU:

15 Los bytes 2-3 de la palabra 1 de una TLE de tipo CPU especifican un entero binario sin signo de 16 bits cuyo valor es la dirección de CPU de la primera CPU en el intervalo de las CPU representado por la máscara de CPU, y cuya presencia está representada por el valor de posición 0 bit en la máscara de CPU. Un origen de dirección de CPU es uniformemente divisible por 64. El valor de un origen de dirección de CPU es el mismo que el almacenado por la instrucción ALMACENAR DIRECCIÓN DE CPU (STAP) cuando es ejecutado en la CPU representada por la posición 0 de bit en la máscara de CPU.

20 Máscara de CPU:

25 Las palabras 2-3 de un TLE de tipo CPU especifican una máscara de 64 bits en la que cada posición de bit representa una CPU. El valor del campo de origen de dirección de CPU más una posición de bit en la máscara de CPU es igual a la dirección de CPU para la CPU correspondiente. Cuando un bit de máscara de CPU es cero, la CPU correspondiente no está representada por la TLE. La CPU tampoco está en la configuración o de otro modo debe estar representada por otra TLE de tipo CPU.

Cuando un bit de máscara de CPU es uno, la CPU correspondiente tiene los valores de atributo de modificador especificados por la TLE, está en la topología de la configuración, y no está presente en ninguna otra TLE de la topología.

30 Así, por ejemplo, si el origen de dirección de la CPU es un valor de 64, y la posición 15 de bit de la máscara de CPU es uno, la CPU 79 está en la configuración y tiene el tipo de CPU, polarización, autorización, y dedicación según ha sido especificado por el TLE.

Ordenación de TLE:

El atributo modificador que se aplica a una TLE de tipo CPU es el tipo de CPU, polarización, autorización y dedicación. La polarización y la autorización (para polarización vertical) son tomadas como un único atributo, aunque con cuatro valores posibles (horizontal, vertical alto, vertical-medio y vertical-bajo).

35 Una única TLE de CPU es suficiente para representar tanto como 64 CPU que tienen todas los mismos valores de atributo modificador.

40 Cuando existen más de 64 CPU, o el margen completo de direcciones de CPU no está cubierto por un único origen de dirección de CPU, y los atributos de modificador son constantes, una TLE de CPU hermana separada es almacenada para cada origen dirección de CPU. Cuando sea necesario, en orden ascendente, de origen de dirección de CPU. Cada una de tales TLE almacenada tiene al menos una CPU representada. La colección de una o más de tales TLE de CPU es llamada conjunto de TLE de CPU.

Cuando existen múltiples tipos de CPU, un conjunto de TLE de CPU es almacenado para cada una, en orden ascendente de tipo CPU.

45 Cuando existen valores múltiples de polarización y autorización, un conjunto de TLE de CPU separado es almacenado para cada uno, en orden descendente de valor y grado de polarización (vertical alto, medio, bajo, y luego horizontal). Cuando están presentes, todos los conjuntos de TLE de CPU de polarización de un tipo de CPU dado, son almacenados antes del siguiente conjunto de TLE de CPU del siguiente tipo CPU.

Cuando existen CPU tanto dedicadas como no dedicadas, un conjunto de TLE de CPU separado es almacenado para

cada una, apareciendo dedicada antes que no dedicada. Todas las TLE son ordenadas suponiendo una primera profundidad transversal donde el orden de clasificación de mayor a menor es como sigue:

1. Tipo CPU
 - a. Valor de tipo CPU menor
 - 5 b. Valor de tipo CPU mayor
2. Polarización-Autorización
 - a. Vertical alta
 - b. Vertical media
 - c. Vertical baja
 - 10 d. Horizontal
3. Dedicación (cuando sea aplicable)
 - a. Dedicada
 - b. No dedicada

La ordenación por origen de dirección de CPU y atributos del modificador de TLE de CPU hermana dentro de un contenedor padre es hecha de acuerdo con la siguiente lista, que va desde el mayor al menor.

1. Conjunto de TLE de CPU de menor valor de tipo CPU, vertical alto, dedicado
2. Conjunto de TLE de CPU de menor valor de tipo CPU, vertical alto, no dedicado
3. Conjunto de TLE de CPU de menor valor de tipo CPU, vertical medio, no dedicado
4. Conjunto de TLE de CPU de menor valor de tipo CPU, vertical bajo, no dedicado
- 20 5. Conjunto de TLE de CPU de menor valor de tipo CPU, horizontal, dedicado
6. Conjunto de TLE de CPU de menor valor de tipo CPU, horizontal, no dedicado
7. Conjunto de TLE de CPU de mayor valor de tipo CPU, vertical alto, dedicado
8. Conjunto de TLE de CPU de mayor valor de tipo CPU, vertical alto, no dedicado
9. Conjunto de TLE de CPU de mayor valor de tipo CPU, vertical medio, no dedicado
- 25 10. Conjunto de TLE de CPU de mayor valor de tipo CPU, vertical bajo, no dedicado
11. Conjunto de TLE de CPU de mayor valor de tipo CPU, horizontal, dedicado
12. Conjunto de TLE de CPU de mayor valor de tipo CPU, horizontal, no dedicado

Otras Reglas de TLE:

1. Una TLE de tipo contenedor está situada a niveles de anidación del orden de 1-5.
- 30 2. Una TLE de tipo contenedor está situada a nivel de anidación 0.
3. El número de TLE de tipo contenedor hermano en una lista de topología o un contenedor padre dado no excede del valor del byte de magnitud (Mag2-6) del nivel de anidación correspondiente a los hermanos.
4. El número de CPU representado por una o más TLE de tipo CPU del contenedor padre no excede del valor del byte de magnitud Mag1.
- 35 5. El contenido de una TLE es definido como sigue:
 - a) Si una TLE es una TLE de tipo contenedor, el contenido es una lista que sigue inmediatamente a la TLE padre, comprendido de una o más TLE hija, y cada TLE hija tiene un nivel de anidación inferior en uno al nivel de anidación

de la TL padre o final de lista de topología.

b) Si una TLE es una TLE de tipo CPU, el contenido es una o más CPU, según se ha identificado por los otros campos de una TLE de CPU.

5 Cuando la primera TLE en un nivel de anidación es una entrada de CPU, el máximo nivel de anidación 0 ha sido alcanzado.

Nota de Programación:

10 Se ha descrito un proceso de examen posible de una lista de topología. Antes de que haya comenzado un examen de una lista de topología, el indicador de TLE actual es inicializado para hacer referencia a la primera TLE o TLE superior en la lista de topología, el indicador del TLE anterior es inicializado a nulo, y a continuación las TLE son examinadas en un orden de superior a inferior.

15 Cuando un examen de lista de topología prosigue, el indicador de TLE actual es hecho avanzar incrementando el indicador de TLE actual por el tamaño de la TLE actual al que indica. Una TLE de tipo contenedor es hecha avanzar añadiendo ocho al indicador de TLE actual. Una TLE de tipo CPU es hecha avanzar añadiendo dieciséis al indicador de TLE actual. El proceso de hacer avanzar el indicador de TLE actual incluye guardar su valor como el indicador de TLE anterior justo antes de que sea incrementado. El examen de TLE no es realizado si la lista de topología no tiene TLE.

El proceso de examen es esquematizado en las siguientes operaciones:

20 1. Si el nivel de anidación de la TLE actual es cero, y el nivel de anidación de la TLE anterior es nulo o uno, la TLE actual representa la primera TLE de tipo CPU de un grupo de una o más TLE de tipo CPU. El programa debería realizar cualquier acción que sea apropiada para cuando un nuevo grupo de una o más CPU es observada en primer lugar. Ir a la operación 5.

25 2. Si el nivel de anidación de la TLE actual es cero, y el nivel de anidación de la TLE anterior es cero, la TLE actual representa una TLE de tipo CPU subsiguiente de un grupo de TLE de tipo CPU que representa hermanas de las CPU previamente observadas en las operaciones 1 ó 2. . El programa debería realizar cualquier acción que sea apropiada para cuando el tamaño de un grupo hermano existente de una o más CPU es incrementado. Ir a la operación 5.

30 3. Si el nivel de anidación de la TLE actual es distinto de cero, y el nivel de anidación de la TLE anterior es cero, la TLE anterior representa una última o única TLE de tipo CPU de un grupo de una o más TLE de tipo CPU. El programa debería realizar cualquier acción que sea apropiada para cuando un grupo existente de una o mas CPU está completado. Ir a la operación 5.

4. Ir a la operación 5.

35 Por eliminación, este sería en caso en el que el nivel de anidación de la TLE actual es distinto de cero, y el nivel de anidación de la TLE anterior es distinto de cero. Si el nivel de anidación de la TLE actual es menor que el nivel de anidación de la TLE anterior, la dirección de la lista de topología transversal se acerca a una TLE de tipo CPU. Si el nivel de anidación de la TLE actual es mayor que el nivel de anidación de la TLE anterior, la dirección de topología transversal se aleja de una TLE de tipo CPU. Las TLE de tipo contenedor están siendo atravesadas conduciendo, bien (1) a otro grupo de TLE de tipo CPU que son un grupo separado en la topología total, o bien (2) al final de la lista de topología. En cualquier caso, no se requiere tratamiento particular más allá de hacer avanzar a la siguiente TLE.

5. Avanzar a la siguiente posición de TLE basado en el tipo de la TLE actual. Si el indicador de TLE actual avanzado es equivalente al final de la lista de topología:

a. No existen más TLE de ningún tipo.

40 b. Si el nivel de anidación de la TLE anterior es cero, el programa debería realizar cualquier acción que sea apropiada para cuando un grupo existente de una o más CPU es completado.

c. El examen está completo.

De otro modo, ir a la operación 1.

45 En una puesta en práctica ejemplar, fig. 16, un sistema de ordenador comprende dos cuadros físicos (Cuadro 1 y Cuadro 2). Cada cuadro contiene dos placas lógicas (Placa 1 y Placa 2), almacenamiento principal (Memoria), y adaptadores de I/O (Canales 1 de I/O y Canales 2 de I/O) para comunicar con dispositivos periféricos y redes. Cada Placa contiene dos módulos multi-chip (Mod 1, Mod 2, Mod 3 y Mod 4). Cada módulo multi-chip contiene dos procesadores (CPU1, CPU2, CPU3, CPU4, CPU5, CPU6, CPU7 y CPU8). Cada módulo contiene también una Memoria Caché de nivel 2 (Caché 1,

Caché 2, Caché 3 y Caché 4). Cada procesador (Unidad de Tratamiento Central o CPU), incluye una Memoria Caché de nivel y una Memoria Tampón de Traducción Anticipada (TLB). Las memorias tampón TLB acceden a información de traslación durante la traducción de dirección dinámica.

5 De acuerdo con el invento, figs. 17, 18 y 19, los componentes del sistema de ordenador son asignados a “contenedores” de acuerdo con su proximidad. Cada módulo es asignado a contenedores más interiores 6, 7, 8 y 9 cuando las CPU de cada módulo están lo más próximas entre sí. Como los módulos están empaquetados en placas, los módulos de las placas respectivas son asignados a contenedores 4 y 5 del nivel siguiente. El siguiente agrupamiento de mayor proximidad es el cuadro. Las placas de un cuadro son asignadas a un contenedor que representa el cuadro (contenedores 2 y 3). El sistema completo está representado por el contenedor 1.

10 Puede verse que dos CPU en el mismo módulo tales como CPU1 y CPU2 están en relación o distancia topológica más próximas entre sí y que ambas residen en un contenedor (contenedor 6) y ningún límite o frontera de contenedor es cruzado cuando solamente están implicadas aquellas CPU. Sin embargo, si la CPU1 y la CPU8 están implicadas, hay 4 límites de contenedor cruzados. La CPU1 en el contenedor 6 está separada por el contenedor 4, 5 y 9 de la CPU8. Por ello, sabiendo la estructura del contenedor, un usuario puede tener una vista de la topología del sistema.

15 Desde luego, en un sistema dividido en particiones lógicamente, las CPU pueden ser compartidas entre sistemas operativos como se ha descrito previamente. Por ello, si a una partición lógica se le han asignado 2 CPU lógicas, cada CPU lógica asignada al 20% de cada una de las tres CPU reales, la partición funcionaría mejor si las 3 CPU reales estuvieran más próximas entre sí, ya que la comunicación entre las CPU y los recursos de CPU (memoria caché y memoria por ejemplo) funcionarían mejor. En nuestro ejemplo, la CPU1 y la CPU2 en una partición experimentaría menos
20 basura de líneas caché en Caché 1 que si las dos CPU fueran CPU1 y CPU8.

En el Ejemplo, una partición es creada incluyendo CPU1, CPU2 y CPU3. Un programa que opera en la partición emite una instrucción STS1 y una tabla de SYSIB 15.1.2 (fig. 10) es devuelta al programa. En este caso, la partición comprende el contenedor 6 y contenedor 7 y por ello hay dos niveles de anidación. Los valores de tabla SYSIB son:

25 El campo MNest es establecido a 4 que indica que la topología incluye 4 niveles de anidación, que es una anidación máxima absoluta para el modelo, y puede o no ser explotado completamente sobre cualquier lista arbitraria de topología, dependiendo de la asignación de recursos a la configuración de invitado que emite STSI, y cómo son asignados los recursos a otras configuraciones de invitado del sistema.

El campo Mag1 es establecido a 2 que indica que 2 CPU están disponibles en el primer nivel más primitivo.

El campo Mag2 es establecido a 2 que indica que 2 contenedores de segundo nivel (Módulo) están disponibles.

30 El campo Mag3 es establecido a 2 que indica que 2 (Placas) de tercer nivel están disponibles.

El campo Mag4 es establecido a 2 que indica que 2 (Cuadros) de cuarto nivel están disponibles.

En nuestro ejemplo, las tres CPU están contenidas en dos módulos de la misma placa, por ello son proporcionadas las siguientes cuatro entradas de TLE:

1. NL = 1; CtnrID= 1 (contenedor 6)

35 2. NL = 0; Tipo CPU = 1 (el tipo de las CPU); CPU Addr Origin = 0; Máscara de CPU
0110.....00 (CPU1 y CPU2 de las CPU accedidas)

3. NL = 1; CtnrID= 2 (contenedor 7)

4. NL = 0; Tipo CPU = 1 (el tipo de las CPU); CPU Addr Origin = 0; Máscara de CPU
00010.....00 (CPU3 de las CPU accedidas)

40 Así, el programa tiene una representación de la topología basada en el contenedor y las TLE de CPU devueltos.

REALIZAR FUNCIÓN DE TOPOLOGÍA (PTF)

Con referencia a la fig. 13, una instrucción de Realizar Función de Topología (PTF) comprende preferiblemente un código de operación y un campo de registro R1 que identifica un primer operando. El contenido del registro general identificado por R1 especifica un código de función en posiciones 56-63 de bit, como se ha ilustrado en la fig. 14.

45 Los códigos de función definidos son los siguientes:

FC	Significado
0	Solicitar polarización horizontal
1	Solicitar polarización vertical
2	Comprobar estado de cambio de topología
5	Los códigos de función indefinidos en el intervalo de 0-255 son reservados para futuras extensiones. A la terminación, si el código de estado 2 está establecido, un código de razón es almacenado en las posiciones 48-55 de bit del registro general R1. Los bits 16-23 y 28-31 de la instrucción son ignorados. Operación de Código de Función 0: La ejecución se completa con el código de condición por cualquiera de las siguientes razones y (códigos de razón):
10	Ninguna razón especificada (0). La configuración de solicitud esta ya polarizada horizontalmente (1). Un cambio de topología está ya en proceso (2). De otro modo, un proceso es iniciado para colocar todas las CPU en la configuración en polarización horizontal. La terminación del proceso es asíncrona con respecto a la ejecución de la instrucción y puede o no ser completada cuando la ejecución de la instrucción se completa.
15	Código de Condición Resultante: 0 Cambio de topología iniciado 1 - 2 solicitud rechazada
20	3 - Operación de Código de Función 1: La ejecución termina con el código de condición 2 por cualquiera de las siguientes razones y (códigos de razón): Ninguna razón especificada (0). La solicitud de configuración esta ya polarizada verticalmente (1).
25	Un cambio de topología está ya en proceso (2). De otro modo, un proceso es iniciado para colocar todas las CPU en la configuración en polarización vertical. La terminación del proceso es asíncrona con respecto a la ejecución de la instrucción y puede o no ser completada cuando la ejecución de la instrucción se completa. Código de Condición Resultante:
30	0 Cambio de topología iniciado 1 - 2 Solicitud rechazada 3 -
35	Operación de Código de Función 2: La condición de informe de cambio de topología pendiente es comprobada, y la instrucción se completa con el establecimiento del código de condición. Código de Condición Resultante: 0 Informe de cambio de topología no pendiente Topología

1 Informe de cambio pendiente

2 -

3 -

5 Un cambio de topología es cualquier alteración tal que el contenido de un SYSIB 15.1.2 sería diferente del contenido del SYSIB 15.1.2 anterior al cambio de topología.

Una condición de informe pendiente de cambio de topología es creada cuando un proceso de cambio de topología se completa. Una condición de informe pendiente de cambio de topología es borrada para la configuración cuando se cumple algo de lo siguiente:

10 Ejecución de REALIZAR FUNCIÓN DE TOPOLOGÍA específica el código 2 de función que se completa con el código 1 de condición.

ALMACENAR INFORMACIÓN DE SISTEMA para SYSIB 15.1.2 es ejecutada satisfactoriamente por cualquier CPU en la configuración.

La reposición del subsistema es realizada.

Condiciones especiales:

15 Si las posiciones 0-55 de bit del registro general R1 son distintas de cero, se reconoce una excepción de especificación. Si se especificado un código de función indefinido, se reconoce una excepción de especificación.

Excepciones de programa:

Operación (El equipo de configuración de topología no está instalado)

Operación privilegiada

20 Especificación

UN ENTORNO EJEMPLAR DE ORDENADOR CENTRAL

25 Como las arquitecturas de servidor final elevado aumentan el número de procesadores físicos y la velocidad del procesador continua mejorando, la "anidación" del procesador necesaria para construir grandes máquinas continúa estando hecha de menores bloques de construcción que son más nodales de naturaleza. Por ejemplo, mientras la memoria caché L2 de un máquina z990 o z9 es una memoria caché totalmente coherente los modelos totalmente popularizados tienen realmente cuatro (4) L2 separadas que están conectadas por un tejido para presentar la apariencia de una única memoria caché L2. La penalidad para salir del nodo para resolver un error de memoria caché continúa aumentando. Por ejemplo, resolver un error de L1 en una L2 remota es más caro que resolverlo en la L2 local. Equivocarse en una CP privada, usualmente en la caché L1 de chip, es caro de poner en marcha y tener que ir todo el camino fuera de la memoria puede parecer una eternidad. El aumento de velocidad de memoria y las conexiones a la misma no está en siendo parejas con los aumentos en la velocidad del procesador. Aunque se podría querer intentar empaquetar todos más juntos "en el chip" o similar, el consumo de potencia y la refrigeración funcionan en sentido contrario.

35 Con la introducción del z990, LPAR resultó conocedora de la topología de la máquina y comenzó a optimizar la asignación de la CP y recursos de almacenamiento a los recursos físicos. Las mejoras de las capacidades para la reoptimización dinámica de asignaciones de recursos de partición lógica fueron introducidas con z9 GA-1 fundamentalmente en soporte de reparación del libro concurrente.

40 El nuevo soporte descrito aquí se dirige al verdadero comienzo de tener software OS zSeries resulta conocedor de la topología de la máquina presentada como una topología de partición lógica, para proporcionar a continuación la distribución de afinidad con respecto a la colocación de la CPU en la estructura del libro CEC.

Se puede pensar del modo en que la LPAR zSeries gestiona las particiones lógicas compartidas, hoy día como que están polarizadas horizontalmente. Es decir, el peso de tratamiento para la partición lógica está dividido igualmente entre todas las CP lógicas en línea en la partición lógica. Este soporte introduce una nueva forma opcional de polarización para gestionar las CP lógicas compartidas de una partición lógica llamada polarización vertical.

45 Cuando una partición lógica elige funcionar en modo vertical, el software emite una nueva instrucción para informar al hipervisor de zSeries de esto y el hipervisor cambiará cómo distribuye la partición lógica.

Dependiendo de la configuración de la partición lógica vertical, los procesados lógicos tendrían una polaridad alta, media o baja. La polaridad describe la cantidad de procesadores lógicos verticales que comparten procesador físico que están autorizados. Los clientes definen los pesos para las particiones lógicas que definen efectivamente la cantidad de ciclos de procesador físico a los que cada partición lógica en una máquina está autorizada.

- 5 La polaridad es medida por la relación de un peso de corriente de partición lógica al número de procesadores lógicos configurados a la partición lógica. Los procesados de alta polaridad tienen cerca del 100% de CPU compartida. Los procesadores de polaridad media comparten >0 a 99% y los procesadores de baja polaridad comparten un 0% (o muy próximo a ello). A las CP lógicas de alta polaridad les será asignado un procesador físico para funcionar muy similar a las CPU dedicadas pero la CP de polaridad alta compartida puede aún dar el recurso físico y permitir que otras CPU compartidas usen sus ciclos sobrantes. La clave aquí resulta entonces que el software vea la topología lógica e intente explotar las CP lógicas muy polarizadas para sus colas de trabajo.

Por ejemplo, un cliente configura un procesador de tres vías con dos particiones lógicas, cada una con dos procesadores lógicos y cada uno con un peso de 50. Si la primera partición lógica estuviera definida en sí misma como vertical, tendría una CP lógica de polaridad alta y una de polaridad media.

- 15 Hay que resaltar que cuando una partición lógica elige funcionar en un modo vertical, la partición lógica completa funciona en modo vertical. Esto incluye la totalidad de sus procesadores secundarios tales como zAAP (IFA) y/o zIIP. Es la responsabilidad del cliente definir pesos a todos los tipos de procesador para estas particiones lógicas que conseguirán el nivel deseado de tratamiento vertical para cada tipo.

Soporte de Topología de Partición Lógica

- 20
- o Establecer infraestructura para una topología nodal de partición lógica.
 - o Hacer cualesquiera cambios necesarios a los algoritmos de asignación nodales de LPAR para particiones horizontales existentes necesarias para proporcionar una topología válida.
 - o Proporcionar simulación de instrucción para el nuevo bloque de información de topología de configuración para la instrucción de Almacenar Información de Sistema (STSI) para proporcionar una topología nodal de partición a la partición lógica.
- 25
- o Examinar cambios en la configuración física o lógica para determinar si es necesario un cambio de topología. Esto puede ocurrir cuando:
 - o Una CP física es añadida o retirada de la configuración.
 - o Una CP lógica es añadida o retirada de la configuración
- 30
- o Una partición lógica es activada
 - o Una partición lógica es desactivada
 - o Los pesos de partición lógica son cambiados desde el HMC/SE
 - o Software de iniciación de cambios de un peso de partición lógica
 - o Una partición lógica es repuesta (cambiar a horizontal)
- 35
- o Una partición lógica cambia a modo ESA/390 (cambiar a horizontal).

ALGORITMOS DEL ENTORNO

- Una topología debe ser asignada a una partición lógica cuando es activada en primer lugar y a continuación cualesquiera cambios en la topología nodal asignada a una partición lógica deben dar como resultado que la partición lógica sea notificada. Los resultados de la topología nodal deben ser conservados en una nueva estructura de datos conveniente para permitir interrogaciones más fáciles por el nuevo tratamiento de STSI así como limitar el tratamiento tanto como sea posible cuando se hacen cambios de configuración. Esta nueva estructura también permite completar el tratamiento de cambio de topología en múltiples operaciones con la puesta en serie requerida para cada etapa sin introducir vistas inconsistentes de la topología a la partición lógica.

Asignación de Topología

- 45 El modo en que es elegida la topología lógica, no es importante para esta exposición. Basta decir que debe hacerse una determinación de cuántos de cada tipo de procesadores lógicos son necesarios y qué nodos o libros necesitan que les

sean asignados. Para una partición de modo vertical, esto significa el cómputo de procesadores verticales altos, verticales medios, y verticales bajos para cada tipo de procesador.

Asignación de valores de Polarización a CP lógicas

5 Una vez que se han determinado los cálculos anteriores, las asignaciones de polarización son hechas desde la dirección de CP lógica en línea más inferior del tipo cp a la más superior en el orden de (1) todos los verticales altos, (2) todos los verticales medios, y (3) todos los verticales bajos. El orden en que esto se hace es arbitrario y son posibles otros órdenes de selección.

Correspondencias de instrucción "Almacenar Información de Sistema"

Añadir 3 estructuras para corresponder con la respuesta 15.1.2:

10 1. Correspondencia para STSI 15.1.2, hacer corresponder el bloque de respuesta con la topología

Dcl	1	syibk1512	char (4096)	based(*),
	3	*	char(2),	
	3	syibk1512_length	fixed(16)	length
	3	syibk1512_mag6	fixed(8)	6th level nest
15	3	syibk1512_mag5	fixed(8)	5th
	3	syibk1512_mag4	fixed(8)	4th
	3	syibk1512_mag3	fixed(8)	3rd
	3	syibk1512_mag2	fixed(8)	2nd, nodes
	3	syibk1512_mag1	fixed(8)	1st ,cpus
20	3	*	char(1),	
	3	syibk1512_mnest	fixed(8)	nesting level
	3	*	char(4),	
	3	syibk1512_topology_list	char(0)	topology list

2. Correspondencia para una TLE de tipo contenedor para STSI 15.1.2

Dcl	1	syibk1512	char (8)	based(*),
25	3	syibknl	fixed(8)	nesting level
	3	*	char(3),	
	3	*	char(1),	
	3	*	char(2),	
30	3	syibk_container	fixed(8);	node id

3. Correspondencia para una TLE de tipo CPU para STSI 15.1.2

Dcl	1	syibk_vcm_cpu	char (16)	based(*),
	3	syibknl2	fixed(8)	nesting level
	3	*	char(3),	
35	3	syibk_ded_polarization	bit(8)	vcm byte
	3	*	bit(5),	

ES 2 379 575 T3

	3	syibk_dedicated	bit(1);	dedicated bit
	3	syibk_polarization	bit(2)	polarization bits
	3	syibk_cputype	fixed(8)	cpu type
	3	syibk_cpuaddrorg	fixed(16)	address origin
5	3	syibk_cpumask	bit(64);cpu mask entry	

TOPBK – Topología de Partición

Un resumen de una topología de partición lógica es conservado actual en este bloque por las rutinas de asignación nodal. Los datos en este bloque están ordenados de tal modo que el tratamiento de STSI puede hacer una pasada de la estructura completa para crear la respuesta de topología de partición lógica al programa, preservando el orden y separación de las entradas de CPU según se requiera por arquitectura.

Consiste de una agrupación tridimensional (nodo, tipo cp, clasificación de polarización) con una máscara de CPU de 64 bits por entrada.

Una segunda área de trabajo, TOP_WORKING es incluida para usar en la actualización de la topología.

DECLARE

```

15  1 TOPBK BASED BDY(DWORD),
      3 TOP_CURRENT,
      5 TOPCPUMASK(1:MAXNODE, /*Cada uno de 1-4 nodos
*/
      0:CPUTMASK, /*índice de tipo CP
20  */
      0:3) /* 4 posibles clasificaciones de topología cuando la partición lógica es vertical.
          Hay solo 2 clasificaciones cuando la partición es horizontal
*/
      BIT(64), /* Máscara de las CPU lógicas que caen en esta clasificación.
25  */
      3 TOP_WORKING CHAR(LEGTH(TOP_CURRENT)); /* Área de trabajo para construir nueva topología
*/

```

TO3PO – Traducción de Topología de Partición

Estas tablas de traducción “constantes” son usadas por las rutinas de asignación nodal que construyen la topología (hecha corresponder por TOPBK) y tratamiento de STSI que lee la topología.

```

30  /*****/

```

```

/* Agrupaciones de traducción de STSI: */

```

```

/*****/

```

DECLARE

```

35  1 TOPVERT(0:3)BIT(8) /* Para particiones verticales, traduce el índice de clasificación anterior a los valores D
(dedicación) y PP (polarización) de arquitectura que han de ser devueltos para este grupo de CP en los datos de STSI
*/

```

```

    STATIC INIT('00000'b || '1'b || PPVH, /* Clasificación 0: Vertical alta Dedicada */
        '00000'b || '0'b || PPVH, /* Clasificación 1: Vertical alta Compartida */
        '00000'b || '0'b || PPVM, /* Clasificación 2: Vertical Media Compartida */
        '00000'b || '0'b || PPVL, /* Clasificación 3: Vertical Baja Compartida */
5   3 *   BIT (5), /* No usada, necesita alineación de byte para agrupaciones de bit */
    3 TOPDPP BIT(3), /* Valores D, PP reales para usar */

1   1 TOPHOR(0:1)BIT(8) /* Para particiones horizontales, traduce el índice de clasificación anterior a los valores D (
dedicación) y PP ( polarización) de arquitectura que han de ser devueltos para este grupo de CP en los datos de STSI.
Obsérvese que solamente las dos primeras clasificaciones pueden ser usadas para una partición horizontal.
10  */

    STATIC INIT('00000'b || '1'b || PPH, /* Clasificación 0: Horizontal Dedicada */
        '00000'b || '0'b || PPH, /* Clasificación 1: Horizontal Compartida */
    3 *   BIT (5), /* No usada, necesita alineación de byte para agrupaciones de bit */
    3 TOPDPP BIT(3), /* Valores D, PP reales para usar */

15  /*****/
    /* Agrupación de traducción de NDI */
    /*****/

1TOPDPP2CLASS(0:7) FIXED /* Usada por las rutinas de asignación nodal para crear la información de topología.
LPDPP es usada como un índice a esta agrupación para determinar que el índice de clasificación debería usar la CP
lógica. Esta agrupación es usada tanto para particiones horizontales como verticales. */
20

    STATIC INIT(1, /* Horizontal, compartida */
        3, /* Vertical baja, compartida */
        2, /* Vertical media, compartida */
        1, /* Vertical alta, compartida */
25  0, /* Horizontal, dedicada */
        0, /* No aplicable */
        0, /* No aplicable */
        0, /* Vertical alta, dedicada */

    3* CHAR(4); /* Forzar para que sea un artículo no simple */

30  Bloque de Procesador Lógico:

Una codificación de 2 bits de polarización de partición puede ser seguida para cada procesador lógico para reflejar su
polarización. Agrupar esto con una indicación de dedicación de 1 bit permite una imagen de polaridad completa para un
procesador lógico en 3 bits:

...
35  3 LPDPP BIT(3), /* Polarización, incluyendo dedicación */
    4 LPDED BIT(1), /* =1, CP lógica es dedicada */
    4 LPPP BIT(1), /* Polarización de Procesador */

```

...

/ Codificaciones para Polarización de Procesador */*

PPH BIT(2)CONSTANT('00'B), */* Polarizada horizontalmente */*

PPVL BIT(2)CONSTANT('01'B), */* Polarizada verticalmente - Baja */*

5 PPVM BIT(2)CONSTANT('10'B), */* Polarizada verticalmente - Media */*

PPVH BIT(2)CONSTANT('11'B), */* Polarizada verticalmente - Alta */*

Actualizar Bloque de Topología

Borrar copia Local de máscara de topología de CPU

DO para cada CP lógica en la partición lógica objetivo

10 IF la CP lógica está en línea THEN

DO

Índice de polaridad = índice de polaridad apropiado para el valor de polarización de las CP lógicas de acuerdo al valor de polarización traducido a la agrupación de índice de polaridad

Copia local de la máscara de topología de CPU (cpu_address, node, cptype, Polarity index) = ON

15 END

END

IF nuevo bloque de topología NOT= bloque de topología actual para partición THEN

Establecer bit de informe de cambio de topología y copiar la nueva topología a la actual.

Simulación de instrucción para STSI 15.1.2

20 Dentro de las correspondencias syibk para el bloque de respuesta de STSI 15.1.2, una TLE de tipo contenedor y una TLE de tipo CPU han sido añadidas. Esencialmente, a los datos deben ser devueltos en contenedor o contenedores siendo la entrada en el nivel más bajo una TLE de tipo CPU. Se puede pensar de esto como una agrupación de agrupaciones basado en como han sido subdivididos o asignados los recursos de partición lógica. Para la realización preferida, cada contenedor es esencialmente un nodo con un nivel de anidación de 1 e incluye unas TLE de tipo CPU cada una de las

25 cuales tiene un nivel de anidación de 0. Las TLE de tipo CPU están ordenadas por el tipo de CPU seguido por su clasificación. Las particiones verticales tienen cuatro clasificaciones (vertical dedicada, vertical alta compartida, vertical media compartida, y vertical baja compartida) y las particiones horizontales tienen dos clasificaciones (dedicada y compartida).

30 Las siguientes operaciones ilustra un caso de utilización para cómo se maneja un STSI 15.1.2 después de que todas las comprobaciones iniciales han validado la entrada de instrucción.

Para la realización actual se han supuesto un máximo de 4 nodos y 64 procesadores.

Se comienza la exploración de topbk, y en una variable local llamada current_node_value se mantiene el valor del índice de nodo en el que se está actualmente. La razón que se necesita es debido a que si todas las máscaras de 64 bits dentro de un nodo son cero, no se necesita crear una TLE de tipo contenedor para ese nodo.

35 Una vez que se ha encontrado la primera entrada distinta de cero dentro de un nodo, en primer lugar se crea una entrada de TLE de tipo contenedor para ese nodo. Dentro de la entrada de la TLE de tipo contenedor el valor de anidación es 1, seguido por 48 bits reservados. Los últimos bits son el ID del nodo que es el índice en topbk del nodo actual que se está tratando. Después de crear la TLE de tipo contenedor, se crea una TLE de tipo CPU para la entrada con una máscara de bit distinto de cero. Dentro de esta entrada, el nivel de anidación es 0, seguido por 24 bits reservados. Los siguientes 8 bits

40 incluyen el bit dedicado y el bit de polarización. Si la partición es actualmente vertical, se llena el valor de polarización y el bit dedicado como sigue:

Clasificación 0 en topbk es vertical dedicada, almacenar un 11 en el PP y 1 en D

Clasificación 1 en topbk es vertical alta compartida, almacenar un 11 en PP y 0 en D

Clasificación 2 en topbk es vertical media compartida, almacenar un 10 en PP y 0 en D

Clasificación 3 en topbk es vertical baja compartida, almacenar un 01 en el PP y 0 en D

Para particiones horizontales, solo son válidas actualmente las clasificaciones 0 y 1. Llénese el bit dedicado y el valor de polarización como sigue:

5 Clasificación 0 en topbk es horizontal dedicada, almacenar un 00 en el PP y 1 en D

Clasificación 1 en topbk es horizontal compartida, almacenar un 00 en PP y 0 en D

El Tipo CPU, el siguiente valor que ha de ser llenado en la TLE de CPU es solo el índice de la segunda agrupación dentro de topcpumask en topbk. (0 – GP, 2 – IFA, 3 – IFL, 4 – ICF, 1 no es usado actualmente).

10 1. El siguiente valor es el origen de dirección de CPU. Este valor es almacenado de manera explícita como 0 ya que 64 es el número máximo de CPU disponibles en la realización actual.

2. El último valor en sybik_vcm_cpu es la máscara de CPU, la máscara de 64 bits distinta de cero almacenada en la agrupación anidada de topcpumask de agrupaciones.

3. Para cada máscara distinta de cero que sigue a la primera máscara de bit distinto de cero dentro de un nodo, crear una TLE de tipo CPU separada e iterar a través de este proceso para los 4 nodos.

15 En una realización, la instrucción PTF podría requerir cambios específicos en la topología distintos de un cambio de polarización, tales cambios incluyen (pero no están limitados a) solicitar que más procesadores invitados sean añadidos a la configuración de invitado, solicitar menos procesadores invitados en la configuración de invitado, solicitar que uno o más procesadores dedicados sean añadidos o quitados de la configuración de invitado, solicitar una polarización específica de procesadores invitados, solicitar que coprocesadores sean añadidos o quitados de la configuración de invitado,
20 cambio temporal de la topología, solicitar un cambio de la topología durante un período de tiempo predeterminado y similares.

Además, el invento no está limitado a topología de procesadores. Puede apreciarse que el componente básico del invento podría aplicarse ventajosamente a componentes distintos de las CPU, incluyendo, pero no estando limitado a coprocesadores, Memorias Caché, TLB, trayectos de datos internos, memorias tampón de trayectos de datos, memoria distribuida y adaptadores de comunicaciones I/O por ejemplo.
25

REIVINDICACIONES

1.- Un método para un cambio de configuración de una topología de una pluralidad de procesadores invitados (CPU invitadas) de una configuración de invitado en un sistema de ordenador anfitrión dividido en particiones lógicas que comprende procesadores anfitriones (CPU anfitrionas), estando el método caracterizado por:

- 5 un procesador invitado de la configuración de invitado que busca (2001) una instrucción de realizar función de topología definida por una arquitectura de ordenador, comprendiendo la instrucción de realizar función de topología un campo de código de operación que especifica la instrucción de realizar función de topología y un campo de registro que especifica un registro general;
- ejecutar la instrucción de realizar función de topología, comprendiendo la ejecución:
- 10 obtener un campo de control de función FC a partir del registro general especificado;
- responder al FC que tiene un primer valor o un segundo valor que especifica una solicitud para un cambio de polarización, es decir, una solicitud para cambiar entre asignar de manera uniforme o no uniforme recursos de tratamiento físicos a los procesadores invitados, solicitar el cambio de polarización especificado de la configuración de la topología de los procesadores invitados de la configuración de invitado;
- 15 responder al cambio de polarización especificado solicitado que es aceptado, cambiar la topología de los procesadores invitados de la configuración de invitado de acuerdo con el cambio de polarización especificado;
- responder al cambio de topología solicitado que está en progreso, establecer (2005) un primer valor de código de condición que indica un cambio de topología iniciado;
- 20 responder al cambio de topología solicitado que es rechazado, establecer (2007) un segundo valor de código de condición que indica que el cambio de topología es rechazado;
- responder al cambio de topología solicitado que es rechazado, establecer un valor de código de razón RC en el registro general;
- 25 responder al FC que tiene un tercer valor que especifica una solicitud para una comprobación de estado de cambio de topología, establecer un valor de código de condición que indica si el informe del cambio de topología está pendiente; y
- completar la ejecución de la instrucción.

2.- El método según la reivindicación 1, en el que ejecutar la instrucción de realizar la función de topología comprende además:

- 30 responder al primer valor FC que especifica una polarización horizontal, iniciar la polarización horizontal de los procesadores invitados de la configuración de ordenador, produciendo por ello una topología actualizada resultante;
- responder al segundo valor FC que especifica una polarización vertical, iniciar la polarización vertical de los procesadores invitados de la configuración de ordenador, produciendo por ello una topología actualizada resultante.

3.- El método según la reivindicación 2, en el que la solicitud de polarización iniciada es asíncrona con la terminación de la ejecución, en que el método comprende además, responder al campo FC que tiene el tercer valor que especifica una comprobación de un cambio de topología, comprobando el estado de terminación del cambio de topología.

4.- El método según la reivindicación 3, en el que la ejecución comprende además:

- 40 responder a un informe de ausencia de cambio en la topología que está pendiente, establecer un código de condición que indica un informe de cambio de topología no pendiente en el que el proceso de cambio de topología está completo; y
- responder a un informe de cambio de topología que está pendiente, establecer un código de condición que indica un informe de cambio de topología pendiente en el que el proceso de cambio de topología no ha sido completado.

45 5.- El método según la reivindicación 2, en el que la polarización horizontal comprende proporcionar un recurso de procesador anfitrión sustancialmente igual a cada recurso de procesador invitado, en el que la polarización vertical

comprende proporcionar sustancialmente más recursos de procesador anfitrión al menos a un procesador invitado de dichos procesadores invitados que al menos a otro procesador invitado de dichos procesadores invitados.

6.- El método según la reivindicación 2, en el que el valor RC especifica un código de razón que consiste de:

5 responder a la configuración que es polarizada como especificada por el código de función antes de la ejecución, indicando el valor RC que la configuración está ya polarizada de acuerdo con el código de función; y

responder a la configuración que trata una polarización incompleta antes de la ejecución, indicando el valor RC un cambio de topología que está ya en proceso.

10 7.- El método según la reivindicación 2, en el que la instrucción de realizar la función de topología definida por la arquitectura del ordenador es buscada y ejecutada por una unidad de tratamiento central de una arquitectura de ordenador alternativa,

en el que el método comprende además interpretar la instrucción de realizar función de topología para identificar una rutina de software predeterminada para emular la operación de la instrucción de realizar función de topología; y

15 en el que ejecutar la instrucción de realizar función de topología comprende ejecutar la rutina de software predeterminada para realizar operaciones del método para ejecutar la instrucción de realizar función de topología.

8.- El método según la reivindicación 2 en el que dicha polarización vertical comprende además aprovisionar dichos procesadores invitados de la configuración de ordenador a un nivel de autorización, siendo el nivel de autorización cualquiera de entre alto, medio y bajo,

20 en el que el nivel alto de autorización comprende asignar la totalidad de un procesador anfitrión a un procesador invitado correspondiente de la configuración de invitado;

en el que el nivel medio de autorización comprende asignar menos de la totalidad de un procesador anfitrión a dicho procesador invitado correspondiente de la configuración de invitado, en el que uno o más procesadores anfitriones son asignados a dicho procesador invitado de la configuración de invitado y

25 en el que el nivel bajo de autorización comprende no asignar parte de ningún procesador anfitrión a dicho procesador invitado correspondiente de la configuración de invitado.

9.- Un sistema que comprende medios adaptados para realizar todas las operaciones del método según cualquier reivindicación de método precedente.

30 10.- Un programa de ordenador que comprende instrucciones para realizar todas las operaciones del método según cualquier reivindicación de método precedente, cuando dicho programa de ordenador es ejecutado en un sistema de ordenador.

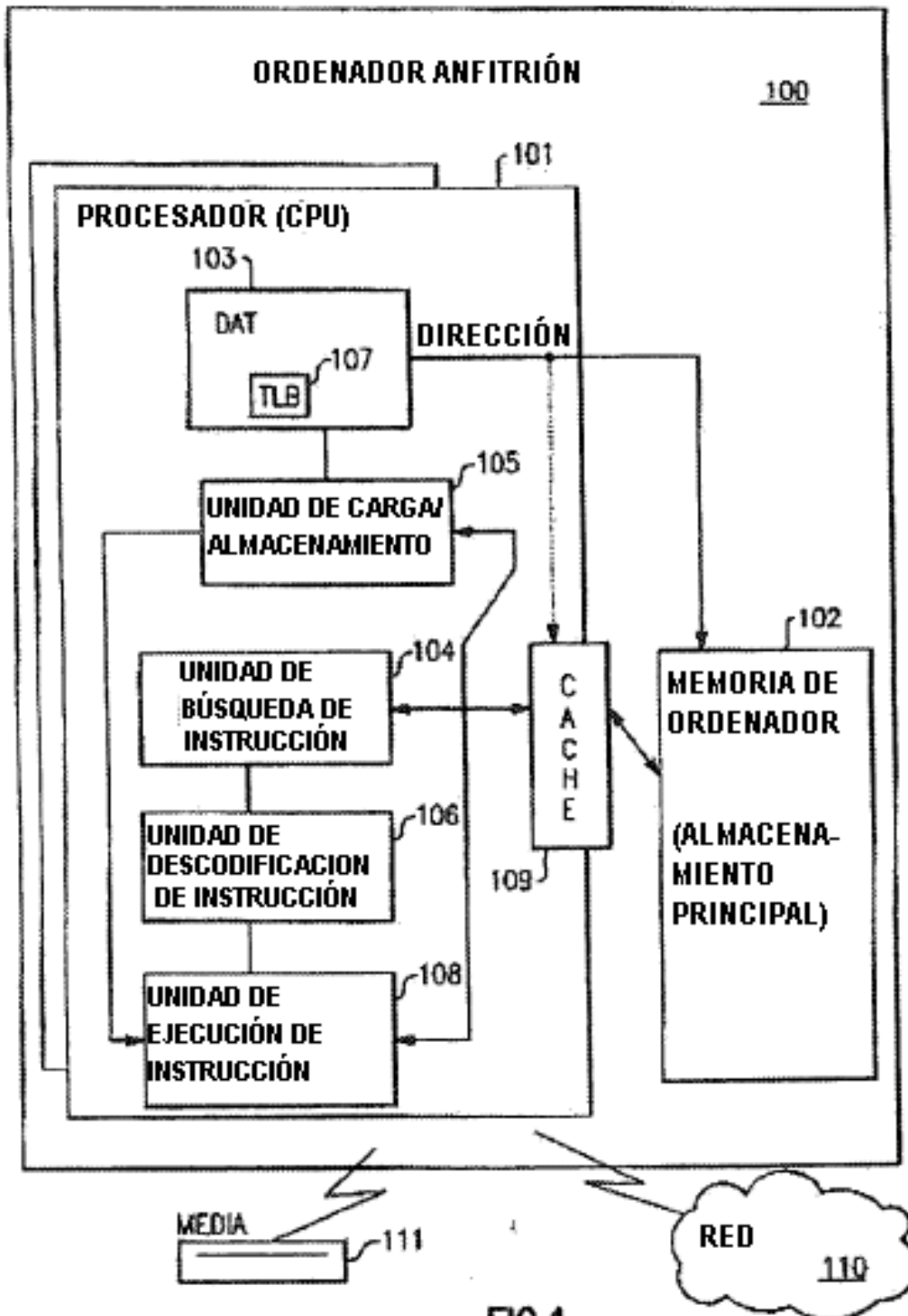


FIG.1

Técnica Anterior

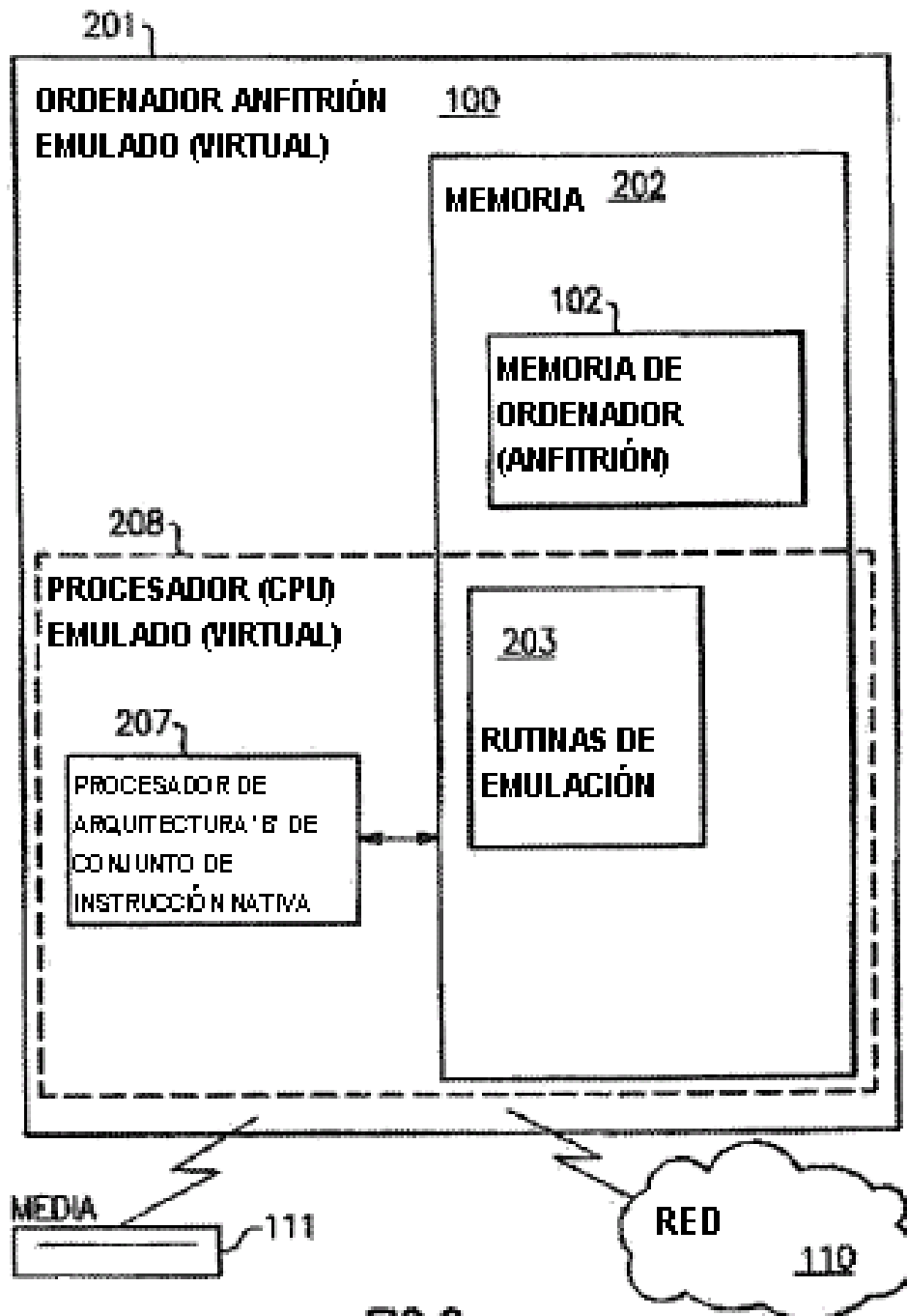


FIG.2

Técnica Anterior

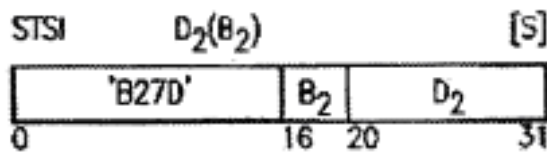
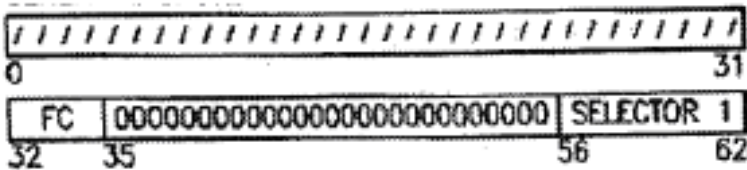


FIG.3

Técnica Anterior

REGISTRO GENERAL 0



REGISTRO GENERAL 1

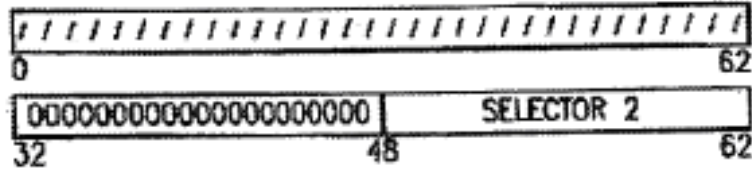


FIG.4

FIG.5

COO. DE FUNC.	SEL. 1	SEL. 2	INFORMACIÓN SOLICITADA
0	-	-	NÚMERO DE NIVEL - CONFIGURACIÓN ACTUAL
1	1	1	CONFIGURACIÓN MÁQUINA-BÁSICA
1	2	1	CPU MÁQUINA-BÁSICA
1	2	2	CPU MÁQUINA-BÁSICA
2	2	1	CPU PARTICIÓN LÓGICA
2	2	2	CPU PARTICIÓN LÓGICA
3	2	2	CPU MÁQUINA-VIRTUAL
15	1	2	INFORMACIÓN DE TOPOLOGÍA DE CONFIGURACIÓN ACTUAL

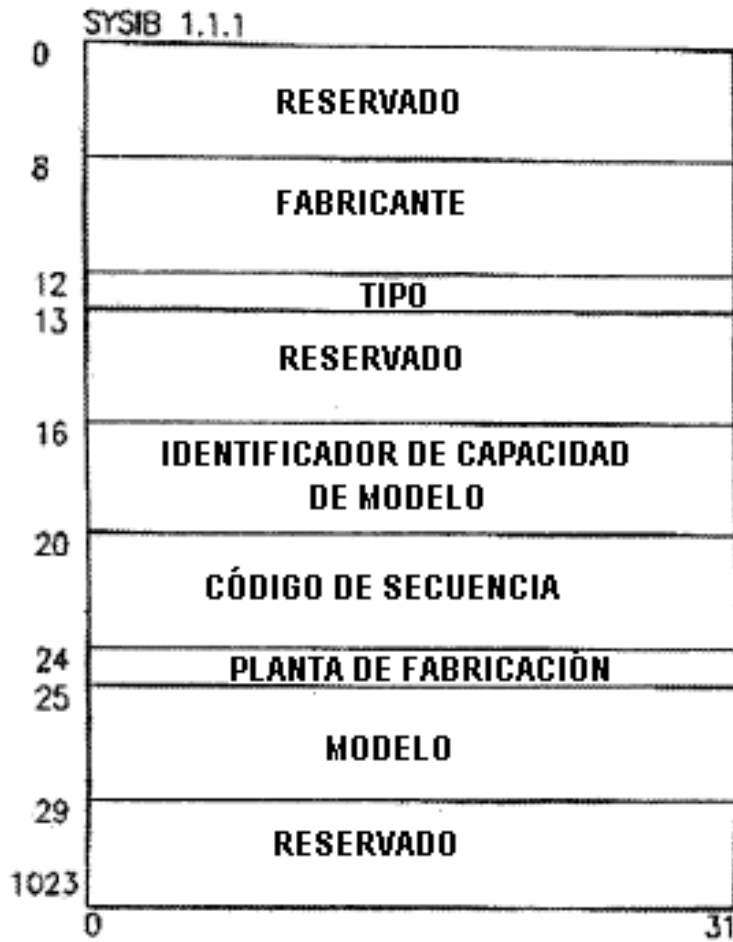


FIG.6
Técnica Anterior

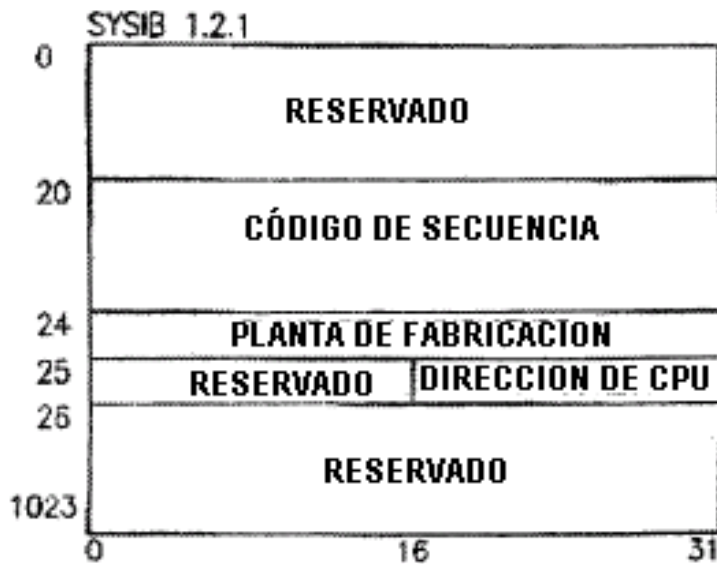


FIG.7
Técnica Anterior

FORMATO-0 SYSIB 1.2.2

0	FORMATO	RESERVADO
1	RESERVADO	
7	CAPACIDAD SECUNDARIA DE CPU	
8	CAPACIDAD DE CPU	
9	CÓMPUTO TOTAL DE CPU	CÓMPUTO DE CPU CONFIGURADO
10	CÓMPUTO DE CPU EN ESPERA	CÓMPUTO DE CPU RESERVADO
11	FACTORES DE AJUSTE DE CAPACIDAD DE CPU MULTITRATAMIENTO	
1023		
	0	8
		16
		31

FIG.8

Técnica Anterior

FORMATO-1 SYSIB 1.2.1

0	FORMATO	RESERVADO	DESPLAZAMIENTO ACC
1	RESERVADO		
7	CAPACIDAD SECUNDARIA DE CPU		
8	CAPACIDAD DE CPU		
9	CÓMPUTO TOTAL DE CPU	CÓMPUTO DE CPU CONFIGURADO	
10	CÓMPUTO DE CPU EN ESPERA	CÓMPUTO DE CPU RESERVADO	
11	FACTORES DE AJUSTE DE CAPACIDAD DE CPU MULTITRATAMIENTO		
N	CAPACIDAD DE CPU ALTERNATIVA		
N+1	FACTORES DE AJUSTE DE CAPACIDAD DE CPU MULTITRATAMIENTO ALTERNATIVA		
1023			
	0	8	16
			31

FIG.9

Técnica Anterior

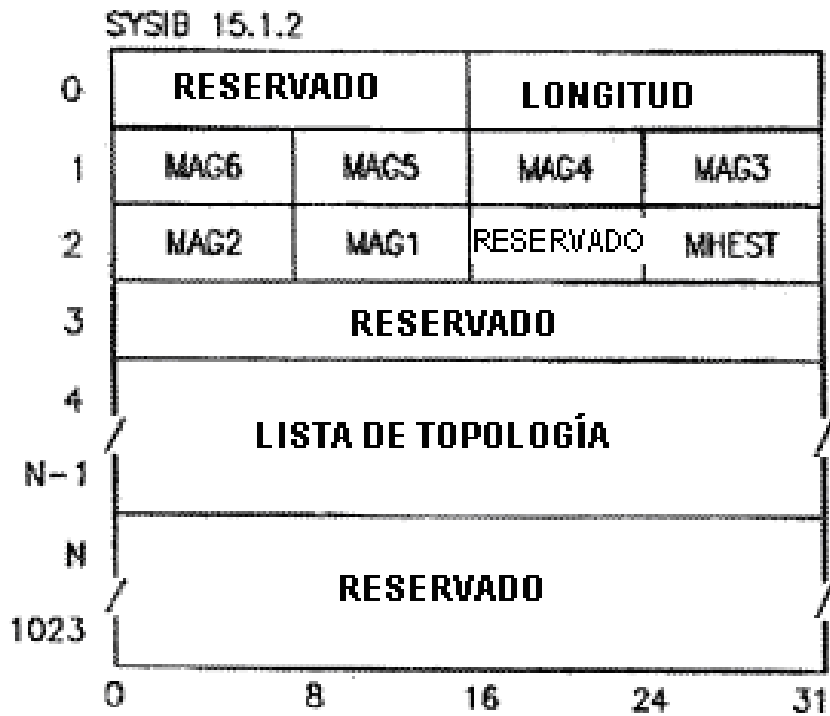


FIG.10

TLE TIPO CONTENEDOR



FIG.11

TLE TIPO CPU

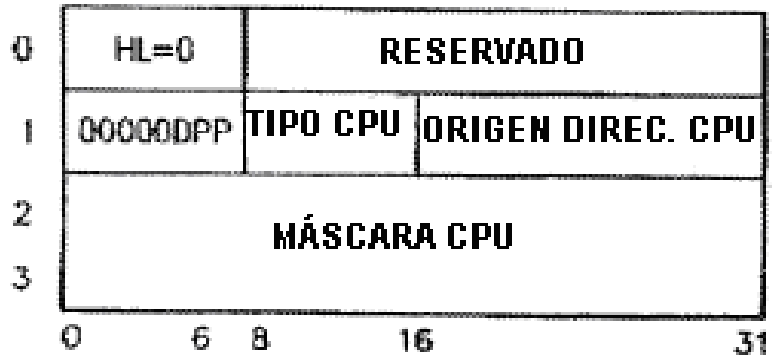
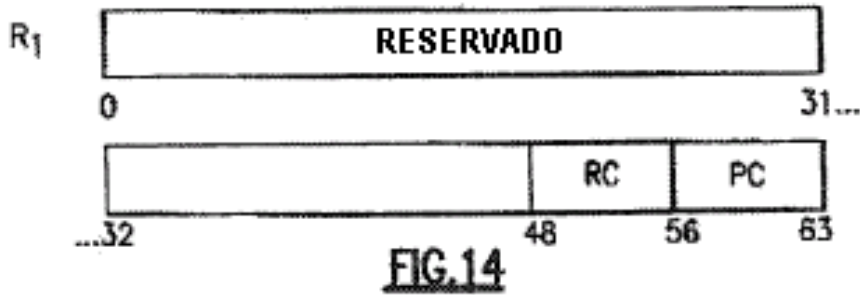
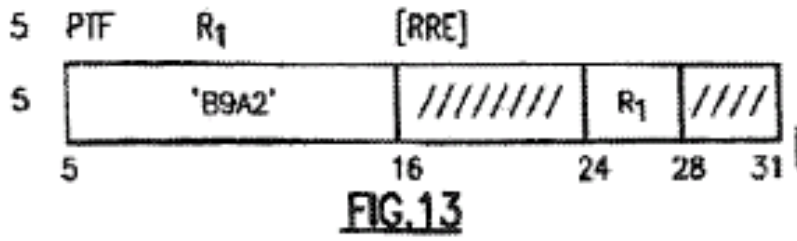


FIG.12



DIVISIÓN EN PARTICIONES

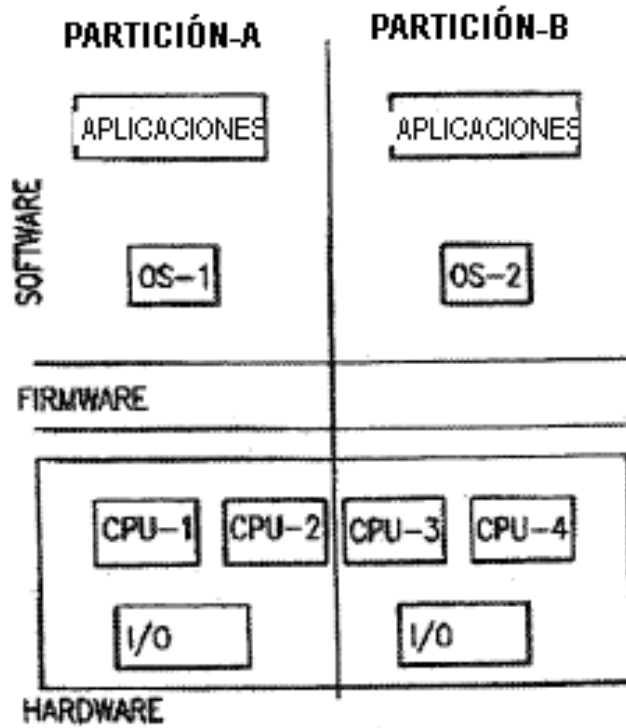


FIG.15A

DIVISIÓN EN PARTICIONES DE SOFTWARE

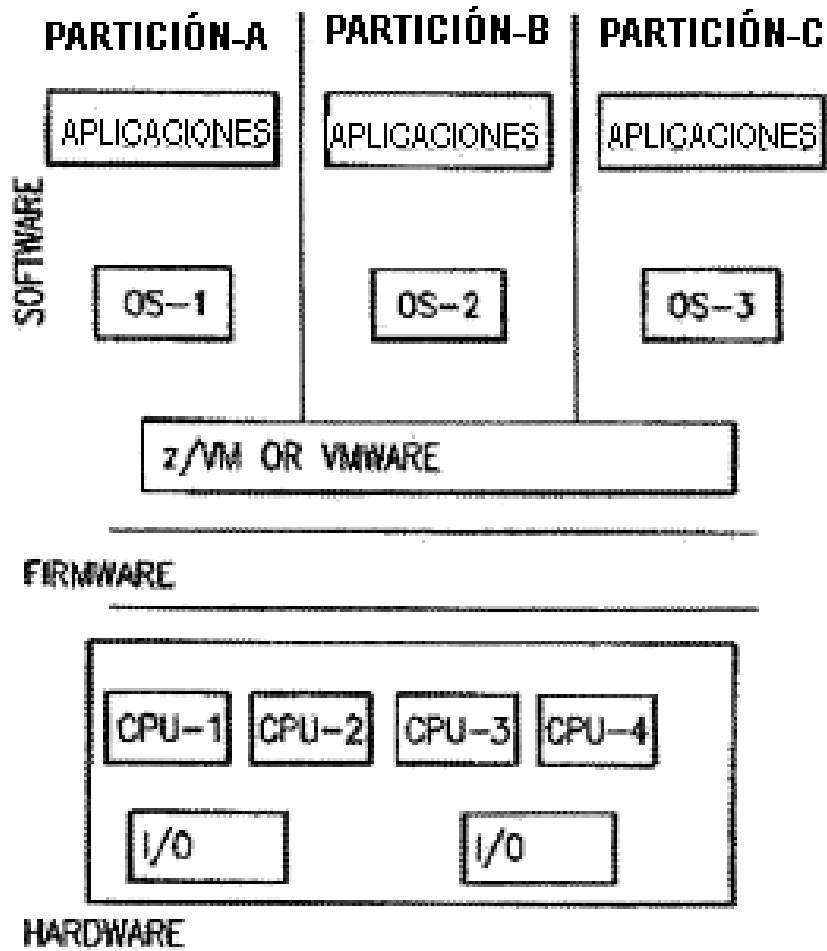


FIG.158

DIVISIÓN EN PARTICIONES LÓGICAS

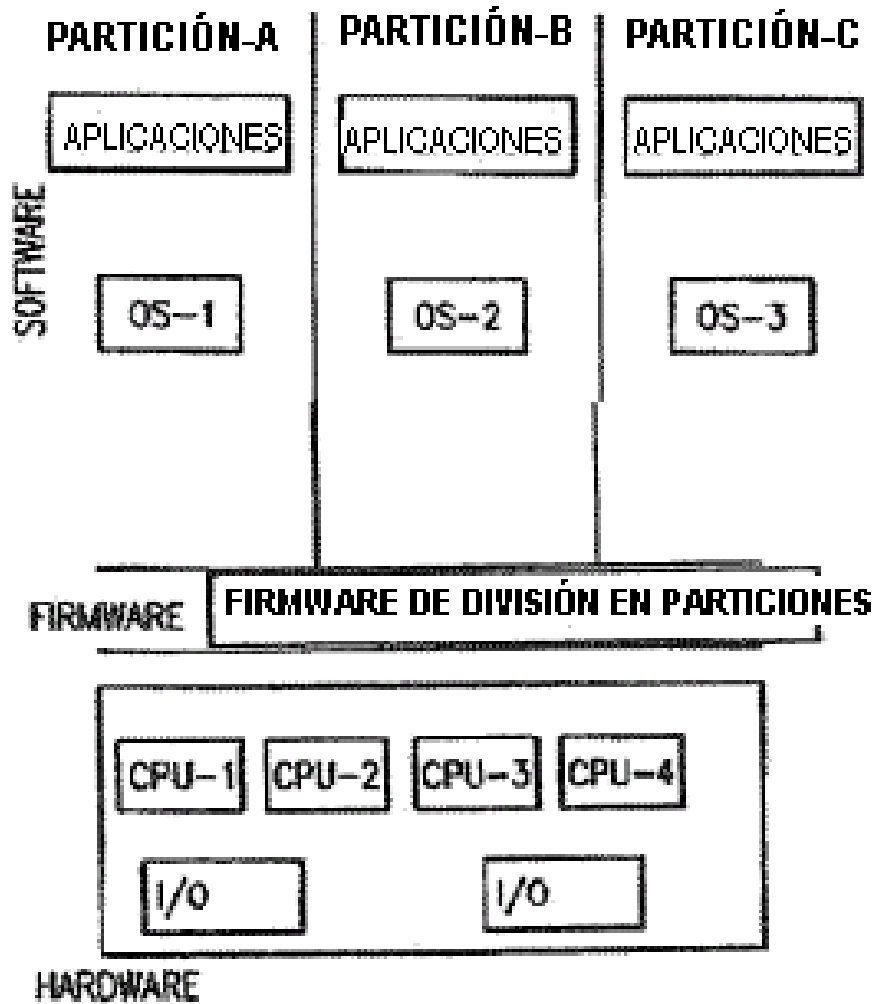


FIG. 15C



FIG.15D

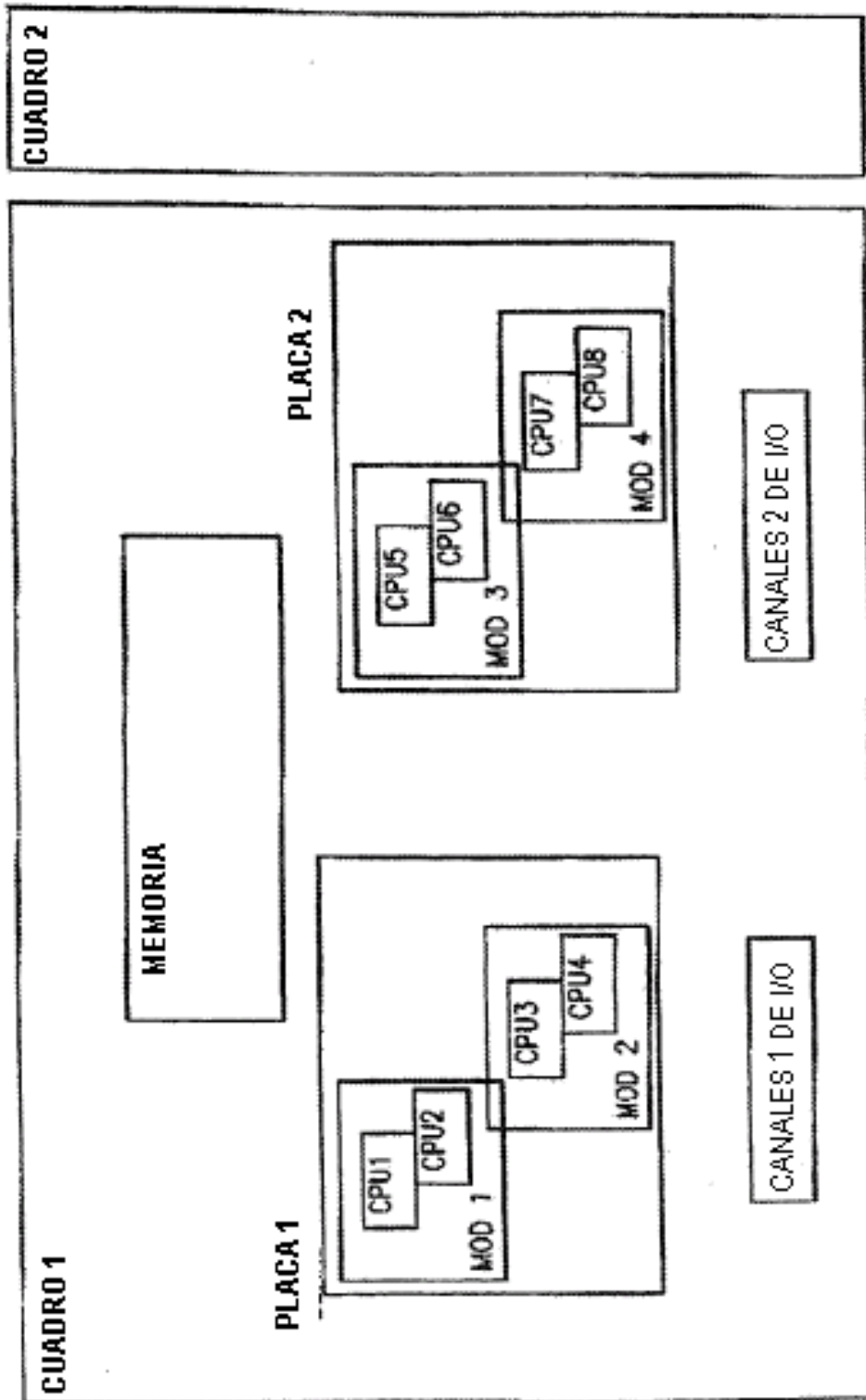


FIG.16

Técnica Anterior

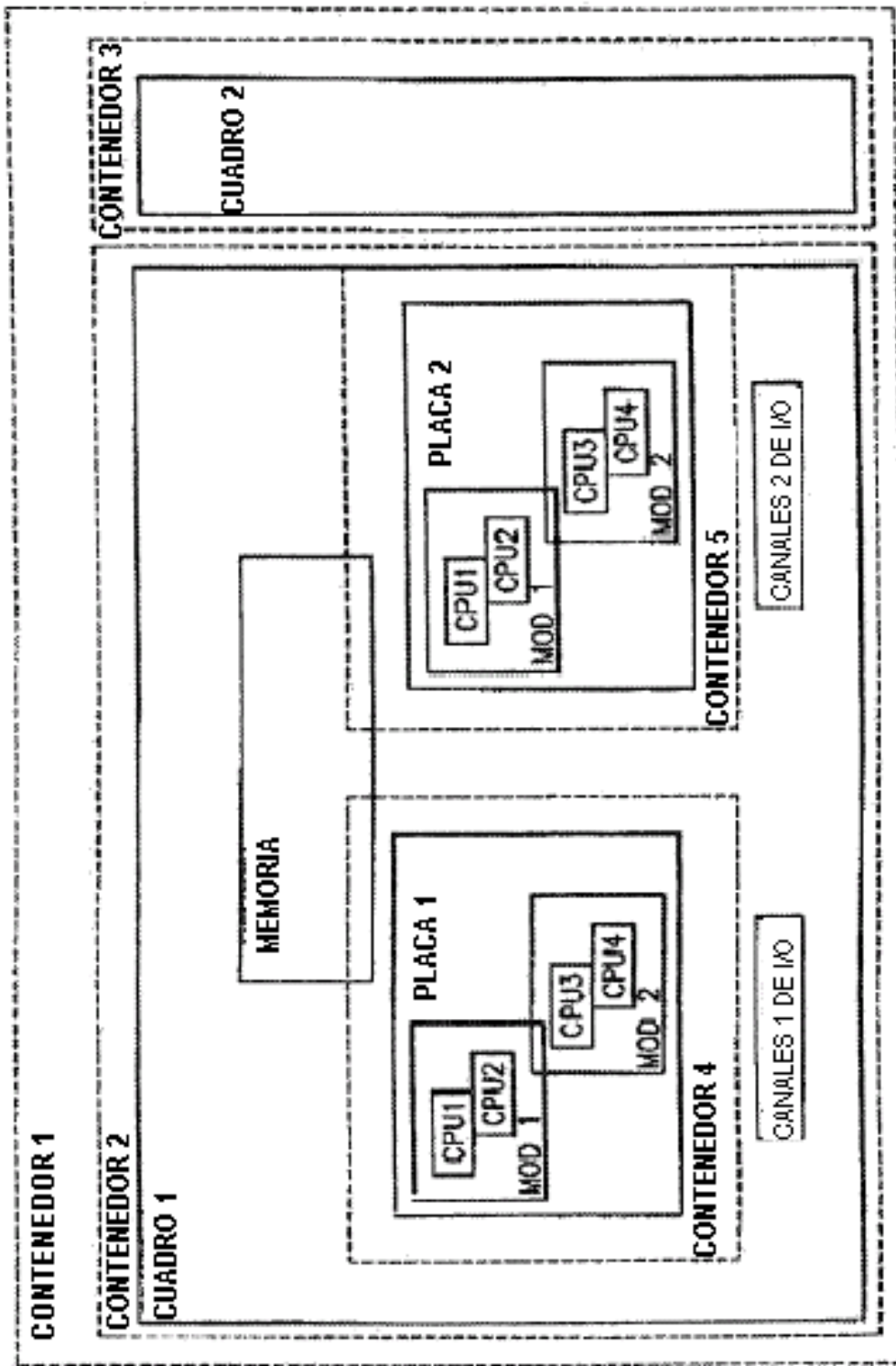


FIG.17

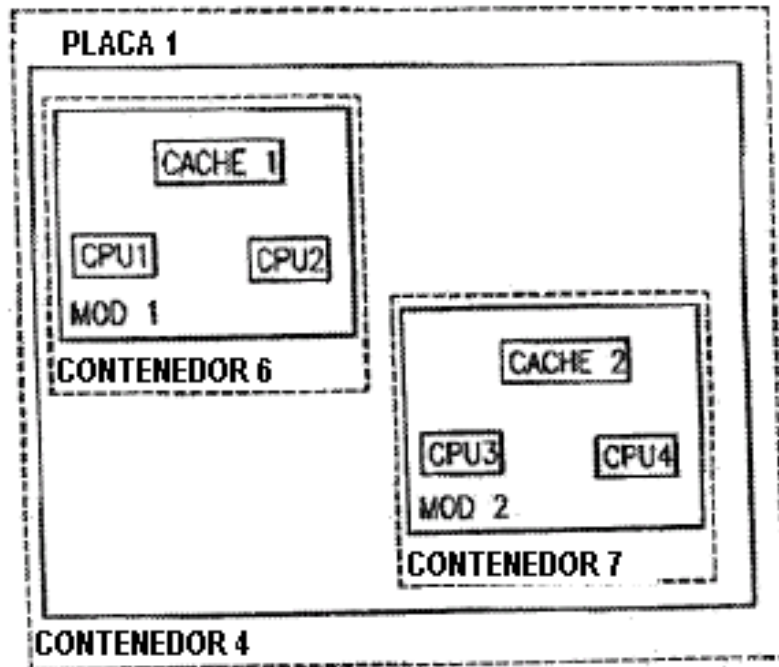


FIG.18

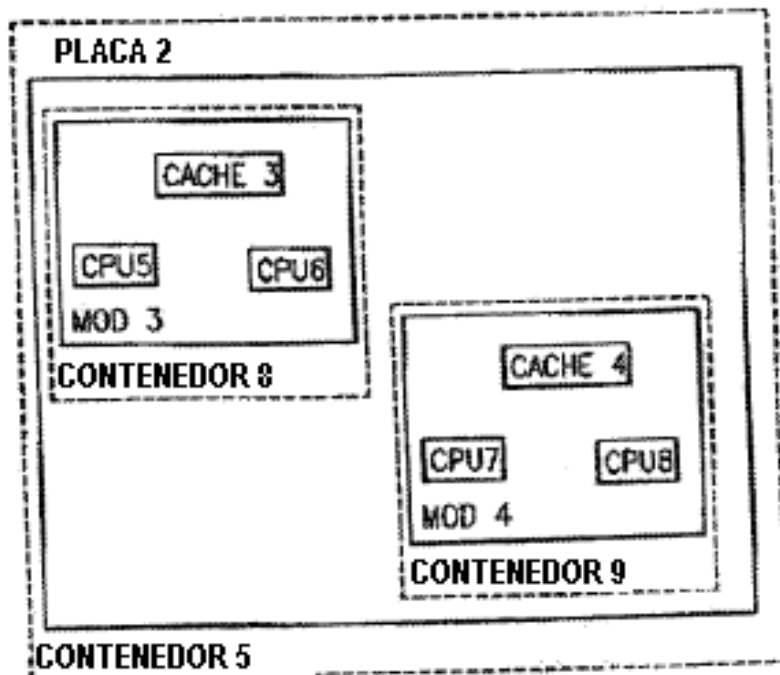


FIG.19

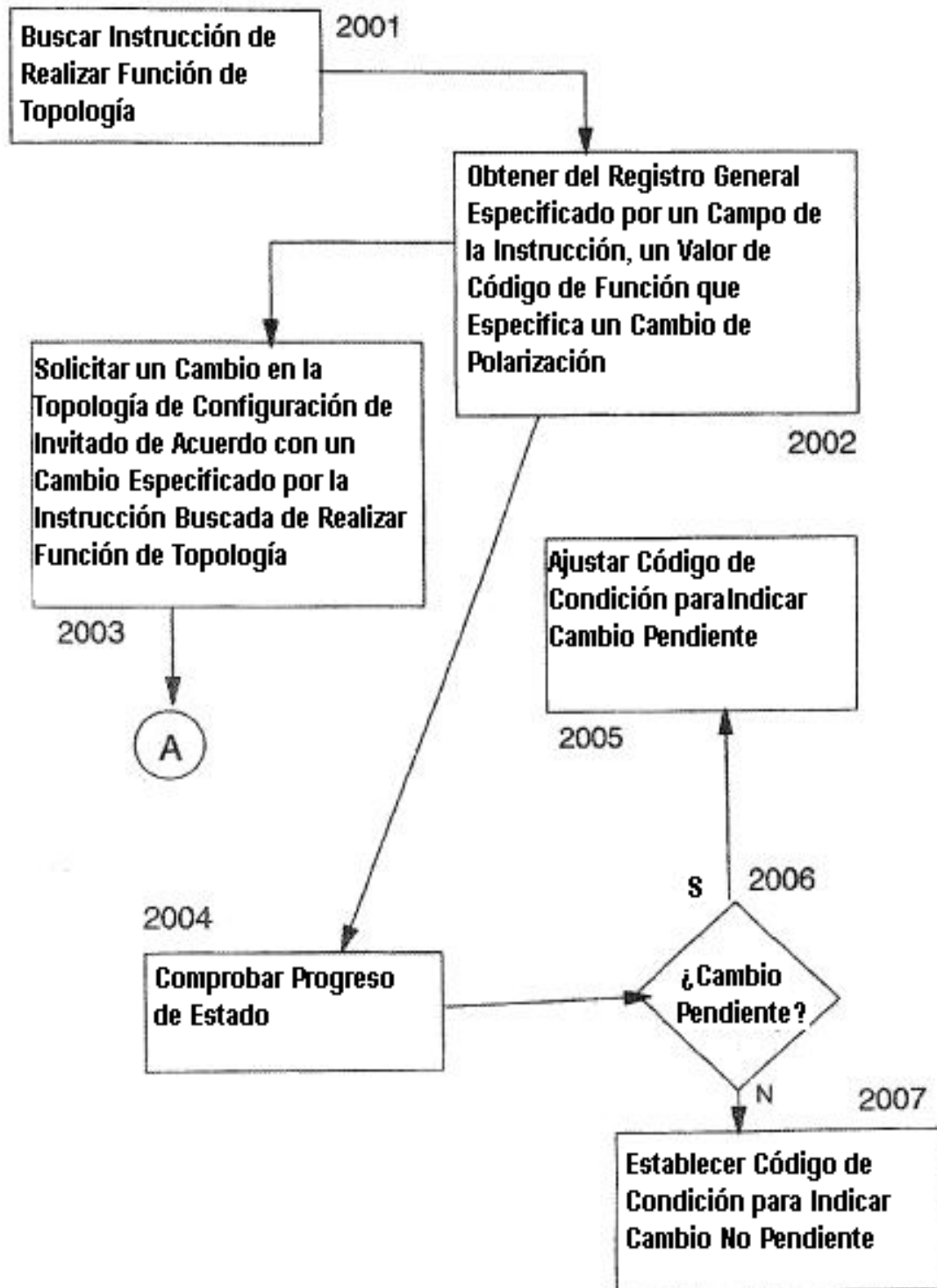


FIG. 20

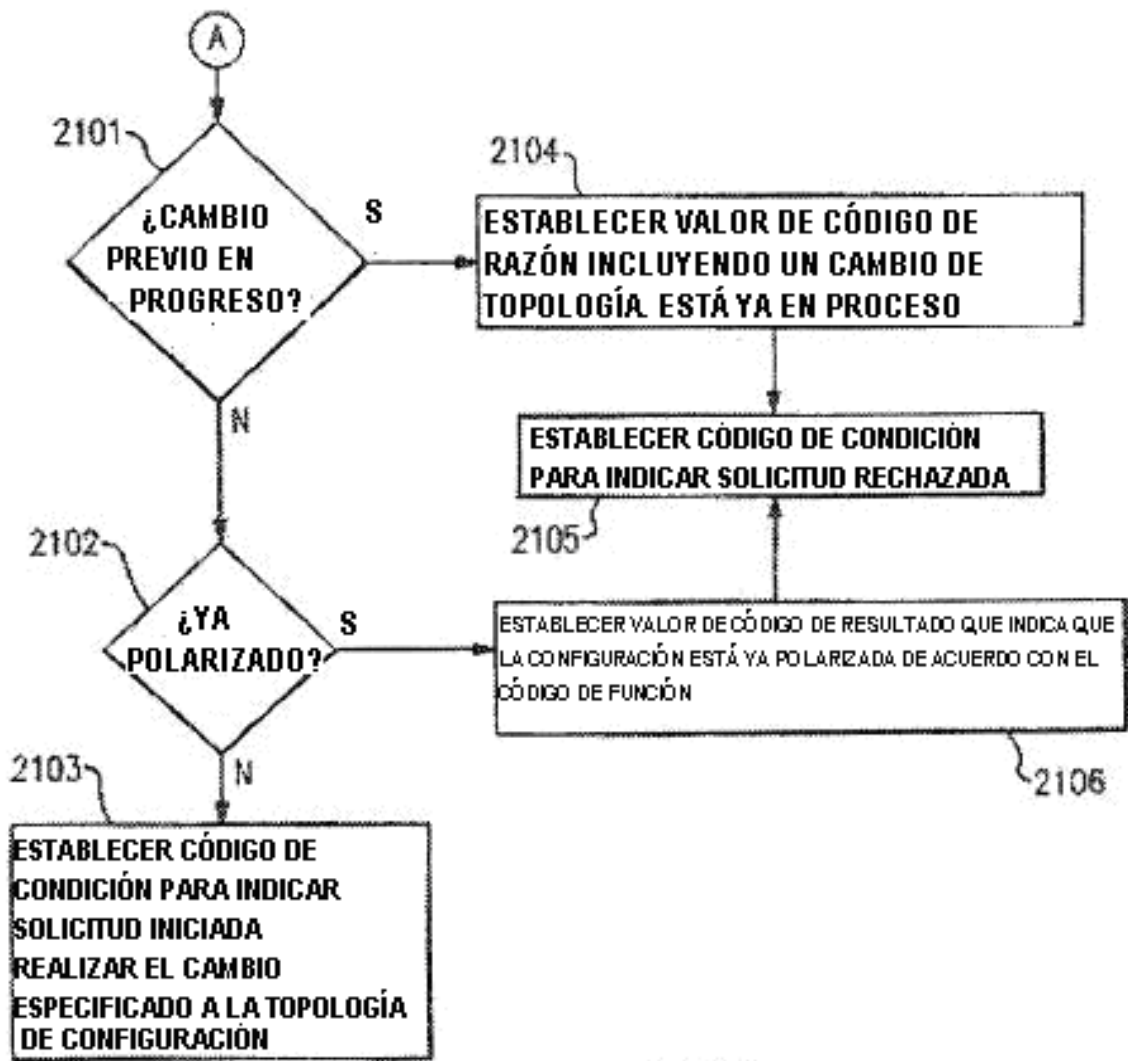


FIG.21